

IMAGE ENHANCEMENT MENGGUNAKAN METODE HIGH-PASS FILTER DAN FAST FOURIER TRANSFORM PADA CITRA TELAPAK TANGAN

SKRIPSI

**MUHAMMAD AKBAR RAMADHAN TANJUNG
0701163110**



**PROGRAM STUDI ILMU KOMPUTER
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SUMATERA UTARA
MEDAN
2021**

IMAGE ENHANCEMENT MENGGUNAKAN METODE HIGH-PASS FILTER DAN FAST FOURIER TRANSFORM PADA CITRA TELAPAK TANGAN

SKRIPSI

Diajukan Untuk Memenuhi Syarat Mencapai Gelar Sarjana Komputer

**MUHAMMAD AKBAR RAMADHAN TANJUNG
0701163110**



**PROGRAM STUDI ILMU KOMPUTER
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SUMATERA UTARA
MEDAN
2021**

PERSETUJUAN SKRIPSI

Hal : Surat Persetujuan Skripsi

Lamp. : -

Kepada Yth.,
Dekan Fakultas Sains Dan Teknologi
UIN Sumatera Utara Medan

Assalamu 'alaikum Wr. Wb.

Setelah membaca, meneliti, memberikan petunjuk dan mengoreksi serta mengatakan perbaikan, maka kami selaku pembimbing berpendapat bahwa skripsi saudara,

Nama	:Muhammad Akbar Ramadhan Tanjung
Nomor Induk Mahasiswa	:0701163110
Program Studi	:Ilmu Komputer
Judul Skripsi	:Image Enhancement Menggunakan Metode High- Pass Filter Dan Fast Fourier Transform Pada Citra Telapak Tangan.

Dapat disetujui untuk segera di *munaqasyahkan*. Atas perhatiannya kami ucapkan terima kasih.

Medan, 25 Maret 2021
12 Sya'ban 1442

Komisi Pembimbing,

Pembimbing I

Pembimbing II

Dr. Mhd. Furqan, S.Si., M.Comp. Sc
NIP. 198008062006041003

Sriani, S.Kom., M.Kom
NIB. 1100000108

SURAT PERNYATAAN KEASLIAN SKRIPSI

Saya yang bertanda tangan di bawah ini,

Nama	:Muhammad Akbar Ramadhan Tanjung
Nomor Induk Mahasiswa	:0701163110
Program Studi	:Ilmu Komputer
Judul Skripsi	:Image Enhancement Menggunakan Metode High- Pass Filter Dan Fast Fourier Transform Pada Citra Telapak Tangan.

Menyatakan bahwa skripsi ini adalah hasil karya saya sendiri, kecuali beberapa kutipan dan ringkasan yang masing-masing telah disebutkan sumbernya. Namun, jika dikemudian hari ditemukan plagiat dalam skripsi ini lalu saya bersedia menerima sanksi pencabutan gelar akademik yang saya dapat dan sanksi lainnya sesuai dengan peraturan yang berlaku.

Medan, 25 Maret 2021

Mhd. Akbar Ramadhan Tanjung
NIM: 0701163110



**KEMENTERIAN AGAMA REPUBLIK INDONESIA
UNIVERSITAS ISLAM NEGERI SUMATERA UTARA MEDAN
FAKULTAS SAINS DAN TEKNOLOGI**

Jl. IAIN No. 1 Medan 20235

Telp. (061) 6615683-6622925, Fax. (061) 6615683

Url: <http://saintek.uinsu.ac.id>, E-mail: saintek@uinsu.ac.id

PENGESAHAN SKRIPSI

Nomor: B.058/ST/ST.V.2/PP.01.1/03/2021

Judul : Image Enhancement Menggunakan Metode High- Pass
Filter Dan Fast Fourier Transform Pada Citra Telapak Tangan.
Nama : Muhammad Akbar Ramadhan Tanjung
NIM : 0701163110
Program Studi : Ilmu Komputer
Fakultas : Sains dan Teknologi

Telah dipertahankan di hadapan Dewan Penguji Skripsi Program Studi Ilmu Komputer Fakultas
Sains dan Teknologi UIN Sumatera Utara Medan dan dinyatakan **LULUS**.

Pada hari/tanggal : Kamis, 25 Maret 2021
Tempat : Ruang Sidang Fakultas Sains dan Teknologi

Tim Ujian Munaqasyah,
Ketua,

Ilka Zufria, S.Kom., M.Kom
NIP. 198506042015031006

Dewan Penguji,

Penguji I,

Penguji II,

Dr. Mhd. Furgan, S.Si., M.Comp.Sc.
NIP. 198008062006041003

Sriani, S.Kom., M.Kom
NIB. 1100000108

Penguji III,

Penguji IV,

Muhammad Ikhsan, S.T., M.Kom
NIP. 198304152011011008

Rakhmat Kurniawan R, S.T., M.Kom
NIP. 198503162015031003

Mengesahkan,
Dekan Fakultas Sains dan Teknologi
UIN Sumatera Utara Medan,

Dr. Mhd. Syahnan, M.A.
NIP. 196609051991031002

ABSTRAK

Telapak tangan sering digunakan sebagai sumber penelitian di bidang sistem biometrik karena mempunyai karakteristik seperti sidik jari. Selain itu, telapak tangan juga mudah didapatkan dan dapat diperoleh dari citra yang memiliki resolusi rendah. Namun, selain itu juga sebuah citra telapak tangan akan dapat mengalami penurunan terhadap kualitasnya. Untuk itu dilakukanlah sebuah tahap yang dikenal dengan *image enhancement* (perbaikan kualitas citra), dimana bidang ini merupakan tahap awal dari pengolahan citra digital. Dalam penelitian ini penggunaan metode dalam *image enhancement* di fokuskan untuk menajamkan citra telapak tangan dengan menggunakan *high pass filter* dan *filter fast fourier transform*, dimana sebelumnya citra tersebut telah diolah dengan menggunakan histogram ekualisasi untuk meningkatkan kontras citra telapak tangan. Setelah dilakukan pengujian terhadap 30 sampel citra telapak tangan. Dengan menilai *error* pada MSE (*Mean Square Error*) dan PSNR (*Peak Signal to Noise Ratio*) dari citra hasil rekonstruksi, hasil pengujian menunjukkan bahwa penggunaan *high pass filter* (HPF) dengan koefisien=1 menghasilkan citra yang lebih baik dimana nilai rata-rata MSE=7,064544 (dB) dan PSNR=40,01314 (dB) daripada menggunakan *high-pass filter* dengan koefisien=0. Sedangkan untuk *filter* pada *fast fourier transform* dengan menggunakan *Ideal High-Pass Filter* (IHPF) mampu menghasilkan citra rekonstruksi yang lebih baik dengan nilai rata-rata MSE=9,354056 (dB) dan PSNR=38,537046 (dB) dari pada menggunakan tapis *butterworth high-pass filter* (BHPF) dan tapis *gaussian high-pass filter* (GHPF).

Kata Kunci: Citra, *Image Enhancement*, *Sharpening*, *High Pass Filter*, *Fast Fourier Transform*

ABSTRACT

Palms are often used as research sources in the biometric system because they have characteristics such as fingerprints. Moreover, palms are also easy to acquire and can be obtained from an image that has a low resolution. Sometimes, however an image of the palm will be subject to drop in quality. For that goes a stage known as image enhancement, Where it's area the first stage of the digital image processing. In this study the use of methods in image enhancement was focused on sharpening a palm image using high pass filters and fast Fourier transform filters, where previously the image has been refined using a histogram equalization to enhance the contrast to the palm image. After testing 30 samples of the palm image. By judging an error on MSE and PSNR from reconstructive images, testing shows that coefficient use of a high pass filter (HPF) with $K=1$ produces a better image where the average value of $MSE = 7.064544$ (dB) and $PSNR=40,01314$ (dB) instead of using a high-pass filter with coefficients $K=0$. Whereas for filters on fast fourier transform (FFT) using the ideal high-pass filter (IHPF) it was able to generate better reconstruction images with equal value of $MSE = 9.354056$ (dB) and $PSNR=38.537046$ (dB) than using a butterworth high-pass filter (BHPF) and a gaussian high-pass filter (GHPF).

Keywords: *Image, Image Enhancement, Sharpening, High Pass Filter, Fast Fourier Transform*

KATA PENGANTAR



Assalamu'alaikum Wr. Wb.

Alhamdulillahirabbil'alamin, puji syukur penulis ucapkan kepada Allah SWT yang mana telah memberikan rahmat dan karunia-Nya. Shalawat dan salam senantiasa tercurahkan kepada junjungan besar Nabi Muhammad SAW. Sehingga penulis dapat menyelesaikan skripsi ini yang berjudul "*Image Enhancement Menggunakan Metode High-Pass Filter Dan Fast Fourier Transform Pada Citra Telapak Tangan*".

Penulisan skripsi ini disusun dalam rangka untuk memenuhi salah satu persyaratan agar memperoleh gelar sarjana Ilmu Komputer pada Fakultas Sains Dan Teknologi Universitas Islam Negeri Sumatera Utara (UINSU) Medan.

Selesainya penyusunan skripsi ini tidak terlepas dari bimbingan, arahan, doa, motivasi serta bantuan dari berbagai pihak sehingga dapat diselesaikan dengan baik. Oleh karena itu, dalam kesempatan ini penulis ingin mengucapkan terima kasih sebesar-besarnya kepada:

1. Bapak Prof. Dr. Syahrin Harahap, MA, selaku Rektor Universitas Islam Negeri Sumatera Utara (UINSU).
2. Bapak Dr. Mhd. Syahnan, M.A, selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Sumatera Utara.
3. Bapak Ilka Zufria, S.Kom., M.Kom selaku Ketua Program Studi Ilmu Komputer Fakultas Sains dan Teknologi
4. Bapak Dr. Mhd. Furqan, S.Si., M.Comp.Sc selaku dosen pembimbing skripsi I yang telah berkontribusi dalam membantu penulis seperti memberikan bimbingan, ide, saran, dan kritiknya kepada penulis selama penulis selama pengerjaan proposal skripsi ini.
5. Ibu Sriani, S.Kom.,M.Kom selaku dosen pembimbing skripsi II yang mana telah berkontribusi dalam membantu penulis seperti memberikan

bimbingan, ide, saran, dan kritiknya kepada penulis selama pengerjaan proposal skripsi ini.

6. Bapak Rakhmat Kurniawan R, ST., M. Kom, selaku Sekretaris Jurusan Ilmu Komputer.
7. Bapak Heri Santoso, M.Kom selaku dosen pembimbing akademik.
8. Seluruh dosen dan staf karyawan, serta seluruh civitas akademika Fakultas Sains Dan Teknologi, terutama Program Studi Ilmu Komputer Universitas Islam Negeri Sumatera Utara (UINSU).
9. Orang tua tercinta, Bapak Dahlan dan Ibu Ernawati yang mana selama ini telah membesarkan dan mendidik penulis dengan penuh kasih sayang, serta tanpa hentinya selalu mendoakan, memotivasi serta telah memberikan dukungan baik secara moril maupun materi.
10. Teman-teman seperjuangan Ilmu Komputer Angkatan 2016, terkhusus Ilmu Komputer-3 yang selalu memberikan motivasi dan arahan yang saat ini sedang berjuang bersama penulis demi meraih gelar sarjana.
11. Kepada saudara-saudara kandung penulis, M. Fahrozi, Agus Syahyuti, Yulia Andani, Aidil Masdhar terima kasih telah memberikan dukungan baik moril maupun materi. Serta terima kasih juga untuk abang dan kakak yaitu Ari Pandjaitan dan Suhaibatul Aslamia yang telah memberikan tempat tinggal dan dukungannya selama penulis menempuh pendidikan di Medan.
12. Serta semua pihak yang telah membantu penulis, yang mana tidak dapat penulis sebutkan satu-persatu.

Penulis berharap semoga Allah SWT memberikan rahmat dan hidayah-Nya kepada mereka semua. Serta dalam penulisan skripsi ini, penulis menyadari bahwa penulisan skripsi ini masih jauh dari kata sempurna. Oleh karena itu, penulis mengharapkan kritikan dan saran yang sifatnya membangun dari berbagai pihak.

Semoga Allah SWT berkenan meridhoi segala apa yang telah kita lakukan dan juga semoga skripsi ini dapat bermanfaat dalam memperkaya ilmu pengetahuan bagi kita semua.

Aamiin Ya Rabbal'alam.

Wassalamua'laikum Wr.Wb.

Medan, 24 Maret 2021
Penulis,

Mhd. Akbar Ramadhan Tanjung

DAFTAR ISI

	Halaman
ABSTRAK	i
ABSTRACT	ii
KATA PENGANTAR	iii
DAFTAR ISI	vi
DAFTAR GAMBAR	ix
DAFTAR TABEL	xi
DAFTAR LAMPIRAN	xii
BAB I PENDAHULUAN.....	13
1.1. Latar Belakang	13
1.2. Rumusan Masalah	15
1.3. Batasan Masalah	15
1.4. Tujuan Penelitian	16
1.5. Manfaat Penelitian	16
BAB II TINJAUAN PUSTAKA	17
2.1. Telapak Tangan (<i>Palmprint</i>)	17
2.2. Defenisi Citra	19
2.3. Jenis-Jenis Citra	21
2.3.1. Citra Analog	21
2.3.2. Citra Digital	21
2.3.3. Jenis Citra Digital	23
2.4. Format File Citra	24
2.4.1. Bitmap (*.bmp)	24
2.5. Pengolahan Citra Digital	25
2.5.1. Sejarah Pegolahan Citra	25
2.5.2. Defenisi Dan Tahapan Pengolahan Citra Digital	27
2.5.3. Operasi-Operasi Pada Pengolahan Citra	30
2.6. Perbaikan Kualitas Citra (<i>Image Enhancement</i>)	33
2.7. Penajaman (<i>Sharpening</i>)	35

2.8. HPF (<i>High-Pass Filter</i>)	39
2.9. FFT (<i>Fast Fourier Transform</i>)	41
2.10. Histogram Ekualisasi (<i>Histogram Equalization</i>)	42
2.11. Pemotongan (<i>Cropping</i>)	43
2.12. <i>Resize</i>	44
2.13. PSNR Dan MSE	44
2.14. <i>Flowchart</i>	45
2.15. Matlab (<i>Matrix Laboratory</i>)	46
2.16. Penelitian Relevan	48
BAB III METODOLOGI PENELITIAN	51
3.1. Lokasi Dan Waktu Penelitian	51
3.1.1. Lokasi Penelitian	51
3.1.2. Waktu Penelitian	51
3.2. Bahan Dan Alat Penelitian	52
3.2.1. Bahan	52
3.2.2. Perangkat Keras (<i>Hardware</i>)	52
3.2.3. Perangkat Lunak (<i>Software</i>)	52
3.3. Prosedur Kerja	52
3.3.1. Perencanaan	52
3.3.2. Teknik Pengumpulan Data	53
3.3.3. Analisis Kebutuhan	54
3.3.4. Perancangan Sistem	55
3.3.5. Pengujian	59
3.3.6. Implementasi	60
BAB IV HASIL DAN PEMBAHASAN.....	61
4.1. Pembahasan	61
4.1.1. Analisis Data	61
4.1.2. Representasi Data	63
4.1.3. Perancangan Sistem	92
4.2. Hasil	95
4.2.1. Pengujian Sistem	96

4.2.2. Hasil Data Uji	106
4.2.3. Implementasi	109
BAB V KESIMPULAN DAN SARAN	111
5.1. Kesimpulan	111
5.2. Saran	112
DAFTAR PUSTAKA	113
LAMPIRAN-LAMPIRAN	

DAFTAR GAMBAR

Gambar	Judul Gambar	Halaman
2.1.	Citra Telapak Tangan (a) dan Ciri Pada Telapak Tangan (b)	19
2.2.	Proses Terbentuknya Citra	20
2.3.	Koordinat Citra Digital	22
2.4.	Tampilan Jenis Citra (a) Citra Warna, (b) Citra <i>Grayscale</i> , dan (c) Citra Biner	24
2.5.	Foto Digital Yang Dikirim Menggunakan Bartlane Cable Picture Transmission System	27
2.6.	Tahapan Dalam Pengolahan Citra Digital	30
2.7.	Teknik Dalam <i>Image Enhancement</i>	34
2.8.	(a) Citra Lena Awal dan (b) Citra Lena Hasil <i>Image Sharpening</i> ..	36
2.9.	Contoh Hasil Konvolusi Pada Tapis HPF (a) Citra Mempunyai Piksel Tepi, (b) Citra Tidak Ada Piksel Tepi	38
2.10.	Citra HPF Dengan Nilai Koefisien = 0	39
2.11.	Transformasi Fourier Dari Domain Spasial ke Domain Frekuensi	40
2.12.	Langkah Dasar <i>Filtering</i> Dalam Domain Frekuensi	41
2.13.	IHPF Dengan $D_0 = 15$	41
2.14.	BHPF Dengan $D_0 = 15$	42
2.15.	GHPF Dengan $D_0 = 15$	42
2.16.	(a) Citra Asli dan (b) Citra Hasil Histogram Ekualisasi	43
2.17.	Tampilan Matlab Versi R2015a	47
3.1.	Tahapan Prosedur Kerja	53
3.2.	Diagram Blok Perencanaan Penelitian	55
3.3.	<i>Flowchart</i> HPF (<i>High-Pass Filter</i>)	57
3.4.	<i>Flowchart</i> FFT (<i>Fast Fourier Transform</i>)	58
4.1.	(a) Citra Sebelum <i>Resize</i> dan (b) Citra <i>Resize</i> 256x256	64
4.2.	(a) Citra Asli 256x256, (b) Citra <i>Grayscale</i> 5x5 dan (c) <i>Region</i> piksel Citra 5x5.....	66

4.3	(a) <i>Region</i> Piksel Citra Histogram Ekualisasi,& (b) Histogram Citra.	67
4.4.	Proses Konvolusi Terhadap Citra	70
4.5.	Representasi Piksel Hasil HPFdengan K=1	74
4.6.	<i>Flowchart</i> Aplikasi <i>Image Enhancement</i>	94
4.7.	Rancangan <i>User Interface</i> Aplikasi <i>Image Enhancement</i>	95
4.8.	Tampilan Aplikasi <i>Image Enhancement</i>	102
4.9.	Tampilan “Buka File Citra”	102
4.10.	Tampilan Citra Setelah di Input ke Sistem	103
4.11.	Tampilan Hasil Histogram Ekualisasi	103
4.12.	Tampilan Filter HPF Beserta Nilai MSE dan PSNR	104
4.13.	Tampilan Citra <i>Filter FFT, Figure Fourier Transform</i> dan Nilai MSE, PSNR	104
4.14.	Tampilan Simpan Citra	105
4.15.	Tampilan Setelah di “Reset All”	105
4.16.	Grafik Nilai MSE.....	107
4.17.	Grafik Nilai PSNR.....	109

DAFTAR TABEL

Tabel	Judul Tabel	Halaman
2.1.	Perbandingan Domain Spasial Dan Frekuensi	35
2.2.	Sifat-Sifat Transformasi Fourier	40
2.3.	Simbol <i>Flowchart</i>	45
2.4.	Penelitian Relevan	51
3.1.	Jadwal Dan Waktu Penelitian	51
4.1.	<i>Database</i> Citra Telapak Tangan	61
4.2.	Nilai Frekuensi dan Distribusi Kumulatif Histogram Ekualisasi	67
4.3.	Hasil <i>Gray Level</i> Histogram Ekualisasi	69
4.4.	Hasil Pengujian <i>Black Box</i> Pada Sistem	96
4.5.	Nilai MSE (dB).....	106
4.6.	Nilai PSNR (dB)	108

DAFTAR LAMPIRAN

Lampiran	Judul Lampiran
1.	Hasil uji citra telapak tangan pada “ <i>Image Enhancement Menggunakan Metode High Pass Filter Dan Fast Fourier Transform Pada Citra Telapak Tangan</i> ”
2.	Alat bantu (<i>tools</i>) yang digunakan pada Matlab GUI untuk “ <i>Image Enhancement Menggunakan Metode High-Pass Filter Dan Fast Fourier Transform Pada Citra Telapak Tangan</i> ”
3.	<i>Source code</i> dari program Matlab “ <i>Image Enhancement Menggunakan Metode High-Pass Filter Dan Fast Fourier Transform Pada Citra Telapak Tangan</i> ”
4.	Kartu Bimbingan Skripsi
5.	Riwayat Hidup

BAB I

PENDAHULUAN

1.1. Latar Belakang

Telapak tangan (*palm*) sering digunakan oleh para ilmuwan sebagai sumber data penelitian khususnya dibidang sistem biometrik. Biometrika ialah ekspansi dari metode dasar identifikasi seseorang dengan menggunakan individualitas alami manusia sebagai basisnya (Setiawan et al., 2013). Telapak tangan adalah salah satu anggota tubuh manusia yang memiliki detail pada bagian permukaan layaknya sidik jari. Tetapi tidak seperti sidik jari, walaupun bentuk telapak tangan sendiri dapat diverifikasi tetapi sebuah telapak tangan manusia tidak *unique* (Prakasa & Herlan, 2017). Kemudahan untuk diperoleh sehingga nyaman bagi pengguna merupakan keunggulan dari pemakaian telapak tangan. Detail pada telapak tangan tersebut mempunyai ciri yang lebih banyak dibandingkan sidik jari dan geometri tangan. Adapun ciri-ciri tersebut meliputi ciri geometri, garis-garis kusut, garis-garis prinsip/utama, serta ada ciri minusi dan titik delta (Dony et al., 2015). Ayat Alquran yang menunjukkan tentang perbedaan antara manusia tersebut dapat di lihat pada QS. Ar-Rum: 22:

وَمِنْ آيَاتِهِ خَلْقُ السَّمُوتِ وَالْأَرْضِ وَاخْتِلَافُ السِّنِّتِكُمْ وَالْوَانِكُمْ إِنَّ فِي ذَلِكَ لَآيَاتٍ لِّلْعَلَمِينَ

Artinya: “Diantara tanda-tanda (kebesaran)-Nya ialah penciptaan langit dan bumi, perbedaan bahasa dan warna kulitmu. Sesungguhnya pada yang demikian itu benar-benar terdapat tanda-tanda (kebesaran Allah) bagi orang-orang yang berilmu.” (QS. Ar-Rum: 22).

Walaupun telapak tangan mempunyai beberapa kelebihan seperti dapat diperoleh dari citra beresolusi rendah (*low resolution*) dan bersifat stabil dalam jangka waktu yang lama. Namun, hasil dari citra telapak tangan terkadang sulit untuk diidentifikasi atau di proses ketahap selanjutnya. Ini disebabkan karena detail atau ciri yang dimiliki pada permukaan telapak tangan maupun citranya yang terkadang memiliki beberapa kendala pada kualitasnya. Permasalahan seperti ini dapat diatasi dengan memperbaiki kualitas pada citra tersebut. Bidang pada ilmu pengolahan citra yang mempunyai fungsi khusus dalam meningkatkan kualitas/mutu citra dikenal dengan *Image Enhancement*. Perbaikan kualitas citra (*image enhancement*) ialah salah satu tahapan awal dalam pengolahan citra (*image processing*), dengan tujuan utama dari *image enhancement* adalah untuk mendapatkan detail lebih banyak dari suatu citra yang tersembunyi dari citra tersebut serta menyoroti informasi yang bermanfaat (G.Khandizod & Deshmukh, 2015). Proses *sharpening* (penajaman) merupakan salah satu bagian dari *image enhancement*, tujuan penerapan *sharpening* pada citra digital agar dapat menghasilkan citra yang lebih mudah untuk diinterpretasikan. Melalui tahap awal inilah kualitas citra diperbaiki sehingga citra yang akan digunakan dapat diolah lebih lanjut.

Ada begitu banyak cara atau metode yang digunakan dalam proses perbaikan kualitas citra (*image enhancement*) khususnya dalam penajaman citra. Antara lain dapat menggunakan *High Pass Filter* (HPF) dan *Fast Fourier Transform* (FFT). Penapis lolos tinggi atau lebih dikenal dengan *High-Pass Filter* (HPF) merupakan dasar sebagian besar metode penajaman citra. Citra yang dipertajam saat kontrasnya ditingkatkan diantara area yang berdekatan dengan sedikit variasi dalam piksel yang cerah maupun gelap. Maka, tepi dapat terdeteksi dengan memanfaatkan filter frekuensi tinggi, sehingga *High-Pass Filter* (HPF) dapat digunakan untuk meningkatkan hasil deteksi citra. *Fast Fourier Transform* (FFT) merupakan sebagai salah satu metode dalam perbaikan kualitas citra yang berbasis transformasi. Dikenal sebagai salah satu metode transformasi yang populer dalam pengolahan citra digital. Dalam prosesnya, metode ini memindahkan

informasi pada citra dari domain spasial kedalam domain frekuensi, yakni dengan merepresentasikan citra spasial sebagai penjumlahan eksponensial kompleks dari berbagai magnituda, frekuensi, serta fasa. Setelah proses enhancement dilakukan, selanjutnya informasi citra dikembalikan ke domain spasial.

Berdasarkan uraian latar belakang diatas, penulis tertarik untuk melakukan penelitian mengenai pengolahan citra dengan menerapkan kedua metode tersebut. Pada tugas akhir ini, penulis menerapkan serta membandingkan metode *high pass filter* (HPF) dan *fast fourier transform* (FFT) dalam proses memperbaiki kualitas citra telapak tangan. Maka, judul dari penelitian ini adalah **“Image Enhancement Menggunakan Metode High-Pass Filter Dan Fast Fourier Transform Pada Citra Telapak Tangan”**.

1.2. Rumusan Masalah

Berdasarkan latar belakang masalah diatas, dirumuskan pokok-pokok permasalahan yang akan menjadi kajian dalam tugas akhir ini, diantaranya:

1. Bagaimana menerapkan metode *High-Pass Filter* (HPF) dan *Fast Fourier Transform* (FFT) dalam memperbaiki kualitas citra telapak tangan?
2. Bagaimana menghasilkan sistem aplikasi *image enhancement* yang dapat digunakan untuk membandingkan dari kualitas citra telapak tangan?

1.3. Batasan Masalah

Untuk menghindari persepsi yang salah serta meluasnya pembahasan yang dapat membuat penelitian ini menyimpang dari ruang lingkup penelitian, maka diperlukan batasan masalah sebagai berikut:

1. Metode yang digunakan dalam penelitian ini adalah *High-Pass Filter* (HPF) dan *Fast Fourier Transform* (FFT).
2. Sistem aplikasi dirancang untuk proses *image enhancement* terhadap citra telapak tangan.

3. Citra yang digunakan adalah citra telapak tangan normal yang sebelumnya sudah di *crop*, terkhusus bagian dalam tidak termasuk jari-jari tangan, dengan format citra *.bmp
4. Ukuran citra yang akan diproses adalah citra hasil *resize* dengan ukuran resolusi 256 x 256 piksel.
5. Dalam menguji kualitas dari hasil rekonstruksi citra dipakailah MSE (*Mean Square Error*) dan PSNR (*Peak Signal to Noise Ratio*).

1.4. Tujuan Penelitian

Untuk menjawab rumusan masalah yang telah di jelaskan, adapun tujuan dari di laksanakan nya penelitian ini adalah sebagai berikut:

1. Untuk menerapkan metode *High-Pass Filter* (HPF) dan *Fast Fourier Transform* (FFT) pada suatu sistem aplikasi.
2. Untuk menghasilkan sistem aplikasi *image enhancement* dengan menerapkan *High-Pass Filter* (HPF) dan *Fast Fourier Transform* (FFT) dalam memperbaiki citra telapak tangan.

1.5. Manfaat Penelitian

Dari penelitian serta penyusunan tugas akhir ini, diharapkan mampu memberikan manfaat antara lain sebagai berikut:

1. Agar memahami penerapan metode *High-Pass Filter* (HPF) dan *Fast Fourier Transform* (FFT) dalam proses *image enhancement*.
2. Mampu memberikan solusi dan menyelesaikan masalah mengenai *image enhancement*, bukan hanya pada citra telapak tangan tetapi pada objek lain misalnya sidik jari.
3. Dapat digunakan untuk penelitian lebih lanjut, misal dalam mengidentifikasi yang terdapat pada suatu sistem aplikasi dalam sistem biometrik.

BAB II

TINJAUAN PUSTAKA

2.1. Telapak Tangan (*Palmprint*)

Dalam ilmu biometrika, telapak tangan merupakan penelitian relatif baru, dimana pertama kali di publikasikan di tahun 1998. Telapak tangan (*palm print*) merupakan bagian fisik manusia yang mempunyai keunikan tersendiri layaknya sidik jari. Sehingga telapak tangan sering digunakan dalam penelitian di bidang biometrika, seperti wajah, sidik jari, iris, suara dan lain sebagainya. Biometrik sendiri adalah suatu sistem yang menghubungkan antara kebutuhan identifikasi/pengenalan identitas seseorang dengan memanfaatkan ciri fisik ataupun perilaku manusia sebagai tanda pengenalnya layaknya seperti kartu identitas diri. Maka, berdasarkan pengertian tersebut biometrika dikelompokkan kedalam 2 jenis, yaitu berdasarkan *physiological characteristics* dan *behavioural characteristics*. Contoh dari *Physiological Characteristics* (ciri fisik) seperti iris, sidik jari, telapak tangan, geometri tangan, suara, retina, DNA dan wajah. *Behavioural characteristics* (ciri perilaku) seperti tanda tangan, *keystroke*, dan pengenalan suara.

Telapak tangan mempunyai tekstur yang detail dan unik, bahkan berbeda dengan telapak tangan kanan dan kiri pada masing-masing orang (Rahman, 2017). Telapak tangan juga memiliki bidang yang jauh lebih luas dibanding dengan sidik jari, sehingga telapak tangan mempunyai ciri-ciri lebih banyak dibanding sidik jari (Dony et al., 2015). Adapun karakteristik adalah sebagai berikut:

1. Ciri Geometri (*Geometry Features*)

Ciri geometri pada telapak tangan merupakan ciri-ciri yang meliputi luas area pada telapak tangan, panjang dan lebarnya, ciri geometri jumlahnya sedikit dan mudah

didapat tetapi mudah juga untuk dipalsukan dikarenakan kemampuan pembeda yang rendah.

2. Ciri Garis-Garis Utama (*Principal-Line Features*)

Ciri yang terdapat pada garis utama cenderung bersifat unik, stabil dikarenakan perubahannya yang lambat dalam periode waktu yang cukup lama. Ciri pada *principal-line* dapat digunakan untuk membedakan masing-masing individu (Renaningtias et al., 2015). Terdapat tiga jenis garis pada ciri utama, yakni garis hati (*heart line*), garis kehidupan (*life line*), dan garis kepala (*head line*).

3. Garis-garis kusut (*wrinkle features*)

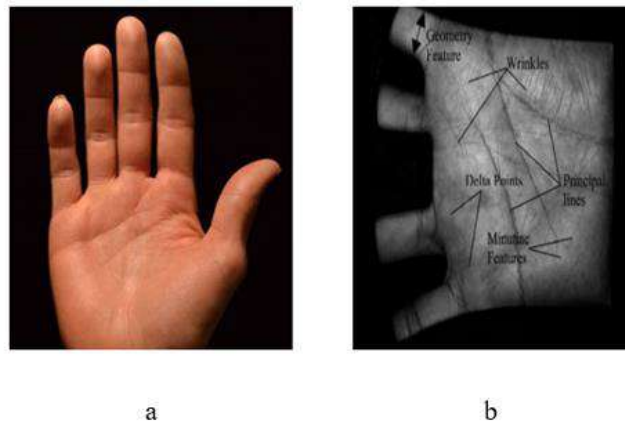
Pada telapak tangan (*palm print*) banyak terdapat garis-garis yang tak beraturan dan terkadang terlihat kurang jelas dan tipis, garis-garis itulah yang dinamakan garis kusut. Sifat dari garis-garis kusut berbeda dengan garis utama, sehingga dapat menghasilkan ciri yang lebih rinci.

4. Ciri Minusi (*Minutiae Features*)

Yang dimaksud sebagai ciri minusi adalah pola-pola berbentuk seperti bukit dan lembah pada bagian permukaan telapak tangan seperti layaknya sidik jari. Ciri minusi didapatkan dari hasil olah citra telapak tangan yang didapat dari resolusi tinggi dan komputasi yang tinggi pula.

5. Ciri Titik Delta (*Delta-Point Features*)

Ada lima (5) daerah delta pada telapak tangan, yakni terdapat pada akar dari tiap jari-jari dan juga bagian luar daerah jari-jari. Ciri titik delta condong ke unik serta stabil, tetapi harus diperoleh dari citra yang beresolusi tinggi untuk menghasilkan ciri ini.



Gambar 2.1 Citra Telapak Tangan (a) dan Ciri Pada Telapak Tangan (b)
(*Semanticscholar.org*)

Adapun keunggulan dari biometrika garis telapak tangan yaitu ciri-ciri pada garis telapak tangan mudah didapatkan dari citra yang beresolusi rendah, permanen, sulit untuk dipalsukan, serta mempunyai bidang yang luas ketimbang sidik jari dan pola-polanya tidak berubah seiring bertambah usia seseorang (stabil) (Purnama & Salsabila, 2013). Kondisi telapak tangan yang dimanfaatkan pada penelitian ini adalah kondisi telapak tangan keadaan normal (tidak kotor, tidak berminyak, tidak ada noda).

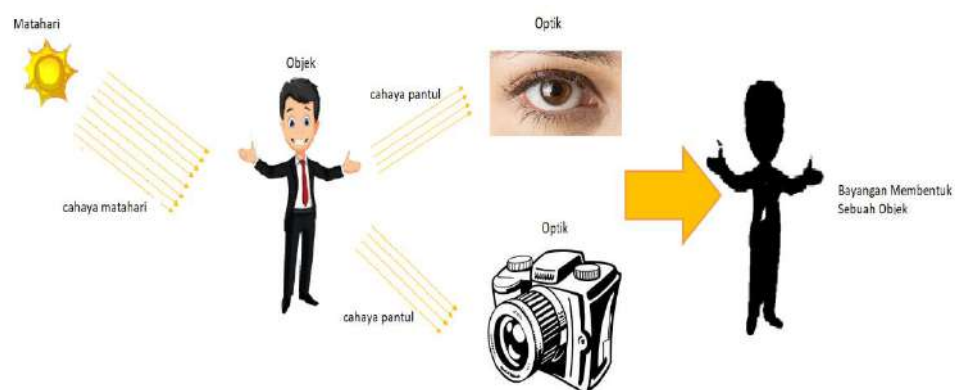
2.2. Defenisi Citra

Sebagai suatu fungsi menerus (*continue*) atas intensitas cahaya terhadap permukaan berdimensi dua ialah sebagai defenisi sebuah citra (Sriani et al., 2017). Sedangkan penejelasan makna lain citra dari kamus *Webster* yang berarti representasi, imitasi dan atau kemiripan dari suatu entitas. Sebagai contohnya foto buah apel mewakili identitas pada buah apel tersebut di depan kamera. Sebuah citra dapat berupa hasil lukisan, fotografi, atau gambaran serta coretan yang ada di kanvas, kertas, dan juga di layar monitor. Dapat dikatakan bahwa citra juga merupakan sebaran dari variasi gelap-terang, redup-cerah, dan/atau warna-warni pada suatu bidang datar. Secara literal, citra (*image*) yakni istilah lain dari gambar, yakni sebuah penjelasan yang ditampilkan dalam bentuk optis/visual. Jadi, citra ialah gambar berdimensi dua yang di dapatkan dari gambar analog berdimensi 2

yang kontinyu menjadi gambar diskrit melewati tahapan *sampling* atau sering disebut digitalisasi.

Representasi spasial dari suatu entitas yang sebenarnya dalam bidang yang berdimensi 2 yang biasa di tulis kedalam posisi *Cartesian* x-y, dan setiap koordinatnya mewakili satu sinyal terkecil dari entitas itu sendiri ialah makna lain citra. Fungsi citra (*image*) yaitu acuan matematik yang biasa dipakai dalam menelaah yang mana semua fungsi analisis diolah untuk membenungkan citra sebagai fungsi bervariasi 2. Citra didapat dari gambaran yang ditangkap oleh kamera maupun alat sensor. Sebagai hasil *output* sistem perekaman data mempunyai sifat optik seperti foto, analog berupa sinyal video pada gambar layar televisi dan digital yang dapat disimpan langsung di pita magnetik. Pada tahun 1983, Simonett dan Joseph mengutarakan pengertian tentang citra, yaitu:

1. Ilustrasi entitas yang dihasilkan melaui pantulan atau pembiasan cahaya yang di pusatkan oleh lensa maupun cermin.
2. ilustrasi rekaman sebuah entitas (seperti foto) yang di buat secara optik, elektro dan optik, optik dan mekanik, ataupun elektronik.



Gambar 2.2 Proses Terbentuknya Citra

Dari gambar diatas, citra di ciptakan dari tangkapan kekuatan cahaya yang di refleksikan oleh objek. Saat sumber sinar datang mencahayai sebuah objek, objek tersebut merefleksikan balik sebagian dari sinar . Hasil pantulan inilah ditangkap oleh peralatan optik, seperti mata, *scanner*, kamera, dan lainnya. Cerminan dari

objek, akan terekam menurut intensitas refleksi sinar yang tertangkap oleh alat-alat optik. Misal sebuah mesin digital yang menangkap refleksi itu, seperti kamera digital, maka citra yang akan diciptakan berupa citra digital. Pada citra digital, kontinuitas intensitas sinar di kuantisasi berdasarkan resolusi dari alat penangkap gambar tersebut.

2.3. Jenis-Jenis Citra

2.3.1. Citra Analog

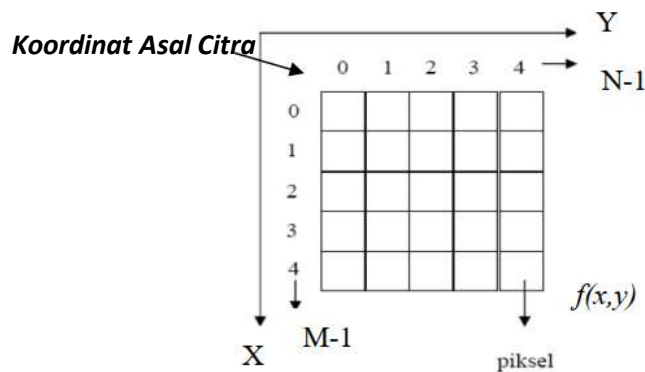
Citra analog yaitu citra ini terbentuk dari sinyal kontinyu. Dengan nilai intensitas cahaya yang mempunyai range antara 0 - ∞ (tak hingga). Citra ini bersifat kontinyu misal seperti plat kendaraan, gambar tercetak, lukisan, gambar dari sinar X, dan lainnya yang terdiri atas sinyal-sinyal frekuensi elektromagnetis yang belum di bedakan umumnya belum bisa di tentukan parameternya (Munir, 2004). Citra ini tidak dapat di representasikan pada komputer, sehingga tidak mampu di olah langsung di komputer. Untuk menangani hal itu, maka citra harus di lakukan pengkonversian sebelum itu dari citra analog ke citra digital. Citra analog di dapat dari peralatan analog misalnya CT *scan*, *webcam*, kamera analog, sensor gelombang pada radar, sensor *ultrasound* pada USG, sensor *rontgen* untuk *thorax*, dan sebagainya.

2.3.2. Citra Digital

Citra digital (*digital image*) yaitu merupakan sebuah citra yang mampu diproses oleh komputer, yang mana citra ini dihasilkan melalui proses digitalisasi dari citra yang bersifat kontinyu. Pada komputer yang tersimpan hanya dalam bentuk angka-angka yang menunjukkan besarnya intensitas setiap piksel. Piksel yakni sebagai elemen sebuah citra dan mempunyai nilai yang menunjukkan intensitas dari warna atau representasi titik terkecil pada sebuah citra grafis. Dikarenakan data yang berbentuk numerik tersebut, sebuah citra digital dapat diolah menggunakan komputer.

Sebuah citra digital tersusun kedalam bentuk *grid*. Kotak-kotak yang membentuk suatu piksel, serta memiliki koordinat (x, y) adalah $f(x, y)$. Dimana

sumbu x merupakan suatu baris (*row*) dan sumbu y adalah kolom (*column*). Masing-masing piksel mempunyai angka yang menampilkan intensitas warna dari piksel itu.



Gambar 2.3 Koordinat Citra Digital (Chahyati, 2004)

Citra digital mampu di representasikan oleh matriks yang dinyatakan kedalam bentuk matriks berukuran $M \times N$. *Picture Element* atau lebih dikenal dengan piksel memiliki dua parameter, yakni koordinat dan intensitas warna. Nilai yang ada pada koordinat (x, y) merupakan besar intensitas warna piksel di titik tersebut. Oleh karena itu, citra digital dapat juga dituliskan kedalam bentuk persamaan matriks berikut:

$$f(x, y) = \begin{bmatrix} f(0,0) & f(0,1) & \cdots & f(0, M-1) \\ f(1,0) & \cdots & \cdots & f(1, M-1) \\ \vdots & \vdots & \ddots & \vdots \\ f(N-1,0) & f(N-1,1) & \cdots & f(N-1, M-1) \end{bmatrix} \dots\dots\dots (2.1)$$

Berdasarkan gambaran dari bentuk persamaan matriks diatas, secara matematis sebuah citra digital dapat ditulis sebagai fungsi intensitas $f(x, y)$, dimana harga x (baris) dan y (kolom) sebagai koordinat posisi dan $f(x, y)$ merupakan sebagai nilai fungsi setiap titik (x, y) yang menunjukkan besar intensitas citra atau tingkat *gray level* atau juga warna dari piksel di titik itu.

Piksel memiliki rentang nilai tertentu dari bernilai terkecil hingga bernilai terbesar. Jangkauan yang di pakaipun tak sama tergantung jenis warnanya. Secara umum, untuk citra 8-bit jangkauannya adalah dari 0 – 255. Citra yang

menggambarkan ini dikelompokkan sebagai citra integer. Berikut ini adalah jenis-jenis citra sesuai nilai pikselnya.

2.3.3. Jenis Citra Digital

Berikut ini adalah jenis-jenis citra digital berdasarkan nilai pikselnya:

1. Citra Biner (*Monokrom*)

Citra biner merupakan citra digital yang memiliki 2 nilai piksel saja yaitu hitam dan putih. Istilah lain dari citra ini adalah citra B&W (*Black & White*) atau sebutan lainnya citra *monokrom*. Citra biner di masing-masing titik piksel bernilai 0 dan 1, hitam direpresentasikan dalam bentuk nilai 0 dan putih adalah bernilai 1. Setiap piksel citra hanya diperlukan media penyimpanan sebesar 1 bit, jadi setiap *byte* dapat menghimpun 8 titik piksel. Citra biner dinyatakan kedalam fungsi berikut ini:

$$I_{Bin}(x, y) = \begin{cases} 0 & I_{BW}(x, y) < T \\ 255 & I_{BW}(x, y) \geq T \end{cases} \dots\dots\dots (2.2)$$

Dan persamaan dalam bentuk *floating point*

$$I_{Bin}(x, y) = \begin{cases} 0 & I_{BW}(x, y) < T \\ 1 & I_{BW}(x, y) \geq T \end{cases} \dots\dots\dots (2.3)$$

Dengan $I_{BW}(x, y)$ adalah nilai titik piksel *gray* titik (x, y) , $I_{Bin}(x, y)$ nilai piksel biner titik (x, y) dan T sebagai nilai *Threshold*.

2. Citra Skala Keabuan (*Grayscale*)

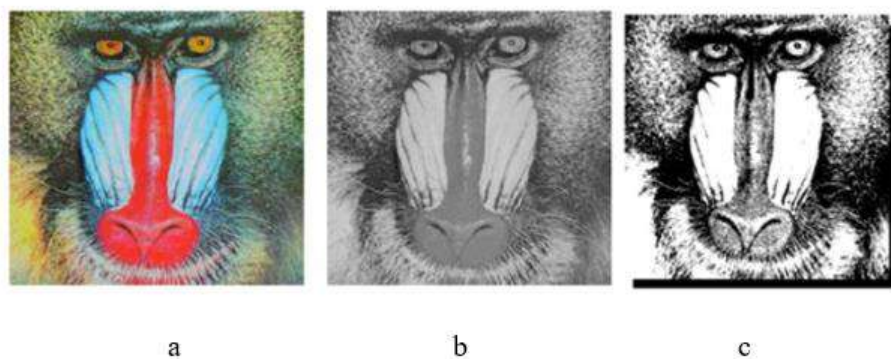
Citra digital ini mempunyai 1 nilai kanal tiap pikselnya, yang berarti nilai bagian *red*, *green* dan *blue* mempunyai warna yang sama, yakni warna hitam, keabuan dan putih. Tingkatan nilai keabuan terdiri dari warna abu dengan macam-macam level dari hitam hingga mencapai putih. Banyaknya kemungkinan minimum dan nilai maksimum dari citra *grayscale* tergantung jumlah bit yang di pakai. Persamaan matematis dari citra *grayscale* adalah seperti berikut:

$$I(x, y) = (I_R(x, y) + I_G(x, y) + I_B(x, y))/3. \dots\dots\dots (2.4)$$

Dimana I_R , I_G , I_B adalah nilai piksel dari *Red* (Merah), *Green* (Hijau), dan *Blue* (Biru).

3. Citra Warna (*True Color*)

Disetiap titiknya memiliki warna yang lebih spesifik dan ialah perpaduan antara 3 warna dasar utama, yaitu merah (*red*), *hijau* (*green*), dan *biru* (*blue*), atau di kenal citra RGB. Nilai intensitas maksimum dari setiap warna adalah 255 (8 bit), dengan warna minimumnya yaitu berwarna putih. Jika, setiap warna mempunyai *range* 0 – 255, maka jumlah keseluruhan warnanya adalah $255^3 = 16.581.375$ (16 K) varian warna berbeda pada suatu citra.



Gambar 2.4 Tampilan Jenis Citra (a) Citra Warna, (b) Citra *Grayscale*, dan (c) Citra Biner (Chahyati, 2004)

Format file pada citra mampu menyatukan dari kualitas citra tersebut, kompabilitas dengan aplikasi serta ukuran filenya. Pada dasarnya, format standar citra yang di gunakan terdiri dari beberapa seperti: *Bitmap* (.bmp), *Tagged Image Format* (.tif, .tiff), *Portable Network Graphics* (.png), *Joint Picture Expert Group* (.jpeg, .jpg) dan lain sebagainya. Format itu di dimanfaatkan dalam menyimpan citra ke suatu file. Pada penelitian penulis kali ini, studi kasus dalam tugas akhir ini akan menerapkan format citra *.bmp.

2.4. Format File Citra

2.4.1. *Bitmap* (.bmp)

Format *Bitmap* (.bmp) dan sering juga disebut *Device Independent Bitmap* (DIB). *Bitmap* merupakan hasil dari representasi citra grafis yang dibentuk dari kumpulan titik-titik yang tersimpan pada memori komputer, dimana tiap nilai tiap titik dimulai dengan satu bit data dalam bentuk gambar hitam putih dan lebih dari 1 bit untuk citra berwarna. Resolusi merupakan kerapatan dari titik-titik yang membentuk citra tersebut. Dimana mampu menampilkan ketajaman gambar saat di tampilkan di layar, dengan ditunjukkan berdasarkan jumlah baris dan kolom

Bitmap adalah sebuah file citra yang tidak dalam bentuk terkompresi, oleh karena itu ukurannya yang relatif besar dari file citra lainnya. Salah satu kelebihan dari penggunaan jenis format *bitmap* ini adalah dapat diakses oleh banyak program pengolahan gambar. Gambar akan terlihat pecah-pecah saat diperbesar atau dapat juga terjadi ketika mencetak dengan resolusi yang lebih kecil dari resolusi citra aslinya dan ini merupakan kelemahan dari format file ini. Contoh format file *bitmap* yang cukup di kenal seperti BMP, TIFF, dan PCX.

2.5. Pengolahan Citra Digital

2.5.1. Sejarah Pengolahan Citra

Peristiwa ini diingat sebagai awal berkembangnya teknik pengolahan citra yaitu dengan berhasilnya pengiriman foto secara digital yang terjadi pada tahun 1921. Saat peristiwa itu terjadi, foto dikirimkan melewati samudra Atlantik dengan menggunakan kabel yang ada dibawah laut dari Kota New York menuju Kota London. Harry G. Bartholomew dan Maynard D. McFarlane dikenal sebagai pencetus sistem pengiriman ini. Sistem ini dikenal dengan Nama “*Bartlane Cable Picture Transmission System*”. Kelebihan dari penggunaan sistem ini adalah reduksi waktu yaitu proses pengiriman foto dimana biasanya memerlukan waktu hingga beberapa minggu dapat dipersingkat menjadi 3 jam saja. Dan kekurangannya adalah berkisar pada teknik pengiriman data yang dilakukan secara digital serta teknik dalam mencetak kembali oleh penerima agar mendapatkan satu resolusi gambar yang baik.

Dalam sistem ini, foto dikirimkan ke dalam bentuk digital dan penerima dengan menggunakan *printer telegraph* diubah menjadi gambar tercetak. Sistem ini pada dasarnya dapat mengkodekan gambar dalam 5 tingkat keabuan. Kemudian berkembang hingga menjadi 15 tingkat keabuan di tahun 1929. Walaupun prosesnya melibatkan gambar digital, namun foto ini tidak dianggap sebagai hasil pengolahan gambar digital dikarenakan tidak menggunakan komputer dalam proses pembuatannya.

Awal tahun 1960, muncul komputer yang mempunyai kemampuan kecepatan proses dan kapasitas memori yang diperlukan untuk melakukan kegiatan pengolahan citra. Pesatnya perkembangan komputer dan program-program tentang angkasa luar mendorong munculnya apa yang disebut pengolahan citra digital. Pada awalnya komputer digunakan untuk meningkatkan kualitas gambar dari hasil foto proyek luar angkasa serta memperbaiki distorsi-distorsi yang didapat saat proses pengiriman.

Pada awalnya pengolahan citra digital hanya dalam mengkonversi citra bentuk analog menjadi digital dan juga memperbaiki kualitas citranya. Dengan seiring perkembangannya, munculah alat-alat pendukung pengolahan citra sehingga membuat penggunaan pengolahan citra digital lebih bervariasi. Dengan menggunakan algoritma pengolahan citra, fungsi sensor penglihatan pada manusia dapat diganti dengan sensor penglihatan buatan atau lebih dikenal dengan kamera. Kecepatan dalam proses komputer yang meningkat pesat, sehingga mampu membuat proses pengolahan citra digital digunakan secara *realtime*. Dan demikian juga perkembangan memori memungkinkan citra analog dapat dikode menjadi citra berwarna yang hampir mendekati warna aslinya.



Gambar 2.5 Foto Digital Yang Dikirim Menggunakan *Bartlane Cable Picture Transmission System* (Gonzalez, 2002)

2.5.2. Defenisi Dan Tahapan Pengolahan Citra Digital

Makna kata pengolahan berdasarkan Kamus Besar Bahasa Indonesia (KBBI) adalah sebagai langkah ataupun proses mengusahakan objek agar berubah atau lebih sempurna. Sedangkan kata citra berarti rupa atau gambar, yang mana dalam hal ini adalah gambar yang diperoleh dengan menggunakan sistem visual. Maka secara keseluruhan dapat diartikan bahwa pengolahan citra ialah sebagai suatu proses atau cara untuk mengusahakan suatu citra agar menjadi citra sesuai yang diinginkan atau lebih sempurna. Berdasarkan makna tersebut, bahwa pengolahan citra itu yaitu suatu proses masukannya berupa citra dan keluarannya berupa citra juga.

Pengolahan citra digital (*digital image processing*) sebagai proses mengolah titik piksel yang ada di dalam citra digital dengan suatu maksud tertentu. Adapun pengertian pengolahan citra digital menurut Gonzalez dan Wood adalah *digital image processing* yaitu proses pengambilan ciri pada suatu citra dengan *input* serta *output* berupa citra (gambar). Pengolahan citra digital (*digital image processing*) digunakan agar bisa melakukan interpretasi pada citra digital. Bagian terpenting dalam pengolahan citra digital adalah memanipulasi serta analisis pada sebuah citra yang dilakukan dengan menggunakan komputer. Umumnya pengolahan citra dibagi menjadi dua jenis kegiatan besar yaitu:

1. Memperbaiki kualitas pada citra, dimana sebuah citra yang diperoleh dapat menampilkan informasi dengan jelas atau dalam arti lain dengan menginterpretasikan citra yang ada, manusia dapat melihat informasi itu.
2. Mengekstraksi informasi dari ciri yang menonjol yang ada pada citra dimana hasilnya berupa informasi citra, dimana manusia mendapatkan data dari ciri pada citra secara numerik dalam makna lain komputer menginterpretasikan informasi melalui besaran-besaran data yang ada dan dapat di pilah dengan jelas, biasanya keperluan data seperti ini diperlukan sebagai informasi untuk pengenalan pola (*pattern recognition*).

Secara umum ada beberapa tahapan atau langkah penting dalam pengolahan citra digital, yang meliputi akuisisi citra, *pre-processing*, segmentasi citra, representasi dan deskripsi, serta pengenalan dan interpretasi.

1. Akuisisi citra dan Pencitraan

Merupakan tahap dimana menentukan data yang di inginkan dan memilih metode perekaman citra digital. Misalnya mengambil, mengumpulkan serta menyiapkan data citra yang akan diambil dengan menggunakan peralatan perekam citra dan memprosesnya sehingga menghasilkan data sesuai yang diinginkan. Pencitraan (*imaging*) adalah proses dimana mengubah informasi atau mentransformasikan citra, dari citra yang tampak atau tidak digital (foto, lukisan, pemandangan, patung dan lain-lain) menjadi citra digital.

2. *Pre-processing*

Tahapan ini dibutuhkan agar menjamin kelancaran ke tahap yang lebih lanjut. Proses penting yang terjadi pada tahap ini antara lain peningkatan kualitas citra, menghilangkan derau (*noise*), transformasi geometri, pengubah kontras dan menentukan bagian-bagian citra yang akan dibutuhkan dalam observasi. Dengan adanya tahap ini, dapat meningkatkan kemungkinan keberhasilan dalam proses pengolahan citra digital selanjutnya.

3. Segmentasi Citra

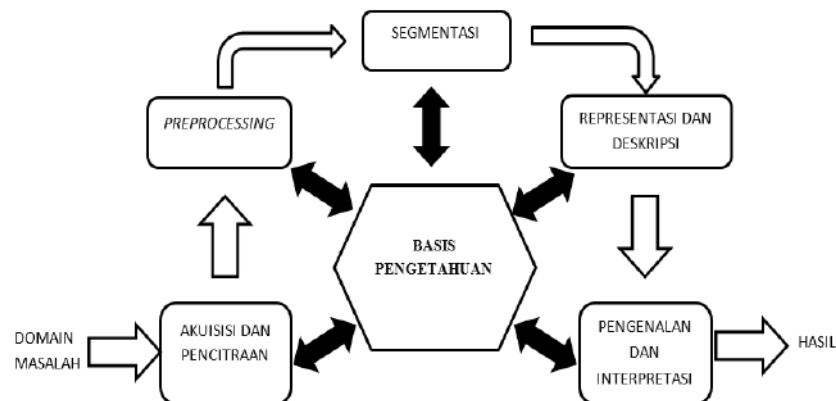
Tahap ini ditujukan sebagai memisahkan citra yang diolah menjadi bagian-bagian penting yang mungkin mengandung informasi/data. Segmentasi citra mempunyai tugas yang paling sulit dalam tahap pengolahan citra digital. Dikarenakan keakuratan proses segmentasi mempengaruhi keberhasilan dalam aplikasi pengenalan objek. Contoh bentuk segmentasi yang sering dilakukan yaitu memisahkan objek/entitas dengan latar belakang citra.

4. Representasi dan Deskripsi

Representasi sebagai suatu proses yang mengacu pada hasil segmentasi kebentuk yang sesuai untuk tahap pengolahan pada komputer. Tahap ini mengikuti keluaran dari proses segmentasi yang berupa data citra mentah yang merupakan salah satu batas suatu wilayah. Data pada citra ini berbentuk runtunan titik-titik koordinat kurva tertutup dengan deskripsi luasan dan parameternya.

5. Pengenalan dan Interpretasi

Tujuan pengenalan untuk memberikan label pada suatu objek yang informasinya telah disiapkan oleh *descriptor*. Dalam tahap ini, pengenalan pola bukan hanya bertujuan agar memperoleh citra dengan kualitas tertentu, tetapi juga harus mengategorikan dari berbagai macam citra. Dari beberapa citra diolah sehingga sebuah citra yang mempunyai ciri-ciri yang sama akan di gabungkan dengan kelompok-kelompok tertentu. Sedangkan tahap interpretasi merupakan bagian yang ditekankan dalam mengenali objek-objek atau memberi penjelasan terhadap kelompok objek/entitas yang dikenali. Basis pengetahuan (*knowledge base*) yang merupakan sebagai basis data atau sumber pengetahuan yang berfungsi untuk mengarahkan operasi dari berbagai macam modul proses dan mengontrol interaksi di antara modul-modul itu. Dan berfungsi sebagai acuan pada tahap pengenalan pola atau *template matching*.



Gambar 2.6 Tahapan Dalam Pengolahan Citra Digital

2.5.3. Operasi-Operasi Pada Pengolahan Citra

Ada banyak operasi pada pengolahan citra digital. Operasi ini dilakukan sebagai memodifikasi sebuah citra menjadi bentuk citra yang berbeda. Operasi-operasi ini di kelompokkan berdasarkan pada tujuan transformasi serta lingkup operasi yang terjadi pada citra tersebut.

Berdasarkan tujuan transformasinya, operasi pengolahan citra digital di kategorikan lagi seperti berikut:

1. Perbaikan Kualitas Citra (*Image Enhancement*)

Merupakan jenis operasi yang bertujuan agar memperbaiki kualitas pada citra dengan memanipulasi kriteria-kriteria yang ada pada citra, sehingga karakteristik khusus yang ada pada citra dapat di tonjolkan. Contohnya seperti perbaikan kontras citra, penapisan derau, penajaman, perbaikan tepi objek, dan pemberian warna semu.

2. Pemugaran Citra (*Image Restoration*)

Tujuan dari operasi ini agar melenyapkan atau meminimalkan cacat pada sebuah citra. Mungkin tujuannya terdengar sama dengan operasi perbaikan kualitas citra, namun bedanya adalah pada operasi ini penyebab terjadinya degradasi pada

citra telah di kenali. Contoh operasi ini adalah penapisan derau, dan menghilangkan *blur*.

3. Pemampatan Citra (*Image Compression*)

Operasi yang di terapkan supaya citra dapat di representasikan kedalam bentuk yang lebih kompak sehingga menyebabkan penggunaan memori lebih kecil. Mempertahankan kualitas citra agar tetap baik merupakan poin terpenting pada operasi ini.

4. Segmentasi Citra (*Image Segmentation*)

Operasi ini sangat berkaitan dengan pengenalan pola (*pattern recognition*) dimana operasi ini memisah citra menjadi bagian -bagian atau segmen dengan kriteria yang telah di tetapkan.

5. Pengorakan Citra (*Image Analysis*)

Bertujuan untuk menghitung besaran kuantitatif dari citra yang nantinya akan menghasilkan deskripsinya. Teknik pada operasi ini yaitu dengan mengekstraksi ciri-ciri tertentu dimana nanti akan membantu pada proses identifikasi objek. Operasi segmentasi terkadang diperlukan dalam melokalisasi objek yang diharapkan dari sekelilingnya. Pendeteksian tepi objek, ekstraksi batas, dan representasi daerah sebagai contoh dalam penerapan operasi ini.

6. Rekonstruksi Citra (*Image Reconstruction*)

Membentuk ulang objek dari aneka citra hasil proyeksi sebagai tujuan dari rekonstruksi citra. Penerapan operasi ini paling banyak dijumpai dalam bidang medis. Seperti hasil foto *rontgen* sinar X dipakai agar citra organ tubuh dari foto di bentuk kembali.

Sedangkan berdasarkan ruang lingkup penerapan operasi ini terhadap suatu citra, operasi-operasi pada pengolahan citra dapat di kelompokkan antara lain sebagai berikut (Munir, 2004):

1. Operasi Titik

Penerapannya dilakukan pada tiap piksel pada sebuah citra yang nilai akhirnya ditentukan berdasarkan nilai piksel tersebut. Operasi ini hanya dilakukan pada piksel tunggal saja. Operasi ini di ulangi pada keseluruhan piksel yang ada di dalam citra. Matematisnya operasi titik dapat dituliskan sebagai berikut:

$$f'(x, y) = O \text{ titik } \{f(x, y)\} \dots\dots\dots (2.5)$$

2. Operasi Lokal

Dilakukan terhadap masing-masing piksel citra yang dimana outputnya di pengaruhi oleh piksel tersebut dan piksel lainnya dalam suatu area tertentu. Dengan kata lain, operasi ini menciptakan citra keluarannya yang intensitas pikselnya tergantung dari intensitas piksel-piksel tetangganya. Adapun contoh penerapan operasi ini seperti operasi konvolusi dalam mendeteksi tepi dan *image smoothing*. Persamaan matematis operasi lokal dapat dinyatakan sebagai berikut:

$$f'(x, y)' = O \text{ lokal } \{f(x_i, y_j); \quad (x_i, y_j) \in N(x, y)\} \dots\dots\dots (2.6)$$

Dimana $N = neighborhood$, adalah piksel-piksel tetangganya yang berada sekitar (x, y) .

3. Operasi Global

Operasi global adalah operasi yang dilakukan pada tiap piksel citra dimana keluarannya ditentukan berdasarkan intensitas dari keseluruhan piksel yang membentuk citra. Contoh operasi ini yaitu penyetaraan histogram untuk meningkatkan kualitas pada citra. Adapun persamaan matematis dari operasi global sebagai berikut:

$$f'(x, y)' = O \text{ global } \{f(x, y)\} \dots\dots\dots (2.7)$$

4. Operasi Objek

Dimana operasi ini dilakukan terhadap objek tertentu pada citra. Yang bertujuan agar mampu mengidentifikasi suatu objek/entitas tersebut, seperti

menghitung rerata intensitas, bentuk, ukuran serta ciri yang lain dari objek tersebut dan operasi ini tergolong sulit.

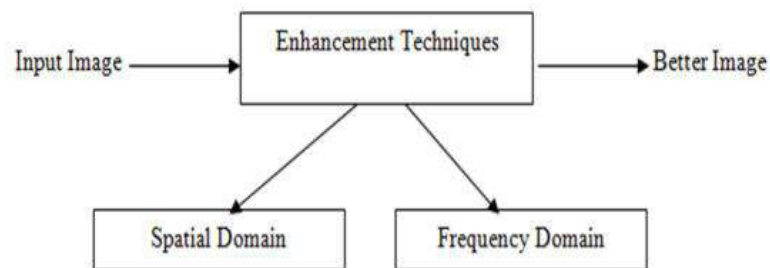
2.6. Perbaikan Kualitas Citra (*Image Enhancement*)

Perbaikan kualitas citra atau dikenal dengan *image enhancement*, dalam pengolahan citra digital dikenal sebagai tahap awal (*pre-processing*). Perbaikan kualitas citra sangat dibutuhkan disaat citra yang akan di olah mengalami penurunan terhadap kualitasnya. Citra yang terlalu terang atau gelap, citra kurang tajam, adanya derau, citra kabur dan lain sebagainya adalah contoh citra yang terdegradasi.

Tahapan-tahapan yang berusaha untuk merekonstruksi atau mengembalikan suatu citra yang terdegradasi sebagai defenisi dari perbaikan citra (Furqan et al., 2020). Tahap untuk mendapatkan citra yang tidak sulit untuk di interpretasikan oleh indera penglihatan manusia merupakan maksud dari proses perbaikan kualitas citra (*image enhancement*). Dengan menonjolkan ciri-ciri tertentu pada citra sehingga citra lebih jelas merupakan cara kerja pada tahap ini. Proses menentukan nilai besaran yang akan diterapkan pada citra adalah bagian tersulit pada tahap ini. Secara matematis, perbaikan kualitas citra (*image enhancement*) dengan kata lain dapat diartikan sebagai proses mentransformasikan citra $f(x, y)$ ke $f'(x, y)$ sehingga ciri-ciri yang dilihat pada $f(x, y)$ lebih di tampilkan (Munir, 2004). Tahapan yang ada dalam perbaikan citra adalah:

1. Pengubahan Kecerahan Gambar (*Image Brightness*)
2. Peregangan Kontras (*Contrast Stretching*)
3. Pengubahan Histogram Citra
4. Pelembutan Citra (*Image Smoothing*)
5. Penajaman Citra (*Image Sharpening*)
6. Pewarnaan Semu (*Pseudocoloring*)
7. Koreksi Geometrik

Dalam perbaikan kualitas citra, ada 2 jenis pendekatan *enhancement* yang sering diterapkan yaitu metode dalam domain spasial dan metode dalam domain frekuensi (Rupa et al., 2018):



Gambar 2.7 Teknik Dalam *Image Enhancement* (Rupa et al., 2018)

1. Domain Spasial (*Spatial Domain*)

Pada domain spasial, teknik dalam pemrosesan gambar berdasarkan dalam metode spasial bekerja secara langsung pada piksel-piksel gambar (Singh et al., 2017). Didalam metode domain spasial memodifikasi nilai-nilai piksel dari citra sesuai aturan yang tergantung dari nilai piksel asli dengan menerapkan operasi lokal ataupun titik (Rupa et al., 2018). Metode dalam pemrosesan citra yang berdomain di ranah spasial dapat dinyatakan sebagai berikut:

$$g(x, y) = T[f(x, y)] \dots \dots \dots (2.8)$$

Dimana $f(x, y)$ adalah citra input, $g(x, y)$ citra output dan T adalah operator terhadap f , dimana dapat beroperasi pada satu piksel (aras titik), piksel yang bertetangga (aras lokal) dan keseluruhan piksel dalam citra (aras *global*).

2. Domain Frekuensi (*Frequency Domain*)

Istilah yang menggambarkan untuk analisis fungsi matematika atau sinyal yang sehubungan dengan frekuensi dan beroperasi langsung terhadap koefisien transformasi pada citra, misalnya seperti transformasi fourier. Pada domain frekuensi, transformasi fourier pada sebuah citra dihitung dan lalu digandakan hasilnya dengan menerapkan *filter* kemudian menerapkan *inverse* transformasi

fouriernya untuk bisa mendapatkan hasil citra asli dari citra yang diperbaiki kualitasnya (*enhancement*) (Rupa et al., 2018).

Dalam teknik domain frekuensi, peningkatan kualitas citra dilakukan dengan cara memanipulasi koefisien transformasi yang memungkinkan operasi pada konten frekuensi citra, maka konten yang berfrekuensi tinggi seperti tepi, batas maupun informasi lainnya dengan mudah bisa dideteksi dan ditingkatkan. Tepian dan transisi tajam pada citra disebabkan oleh frekuensi tinggi dalam transformasi fourier sedangkan bagian halus yang muncul pada citra disebabkan oleh frekuensi rendah (Singh et al., 2017). Contoh metode pada domain frekuensi seperti tapis rendah, tapis tinggi, dan homomorfik.

Tabel 2.1 Perbandingan Domain Spasial dan Frekuensi (Prasetyo, 2011)

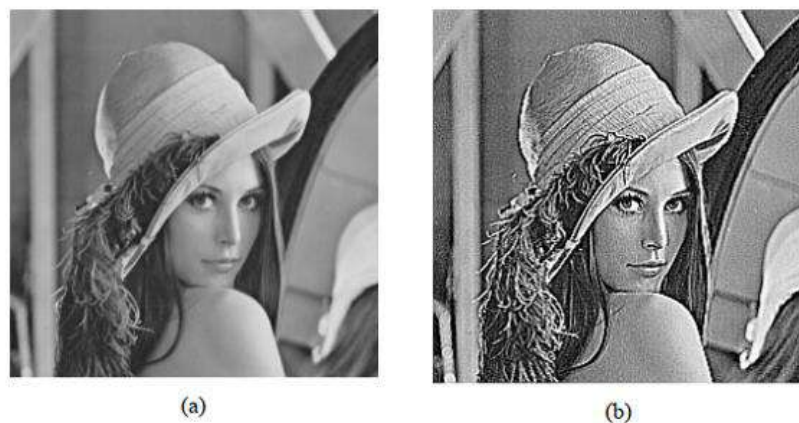
No.	Domain Spasial	Domain Frekuensi
1	Konsep koordinat baris dan kolom	Konsep frekuensi, maksudnya yaitu perubahan intensitas dari piksel ke piksel (perubahan dari frekuensi rendah dan tinggi).
2	Pemrosesan yang terjadi secara piksel per piksel	Pemrosesannya disesuaikan berdasarkan pemilihan frekuensi-frekuensi yang akan dilakukan filter atau tidak.
3	Komputasi/pemrosesannya terbilang lama (terutama jika citra dengan ukuran spasial yang tinggi).	Komputasi relatif cepat (terutama jika citra memiliki ukuran spasial tinggi).

2.7. Penajaman (*Sharpening*)

Sharpening atau penajaman merupakan operasi dengan tujuan agar memperjelas tepi-tepi pada suatu objek dalam citra digital. Dalam hal ini piksel yang berada di tepi objek akan di tampilkan lebih jelas (*highlight*), sedangkan pada

piksel yang bukan bagian dari tepi akan dibuat gelap atau hitam. Operasi penajaman citra adalah kebalikan dari pelembutan citra atau *image smoothing*. Fungsi *sharpening* (penajaman) sering kali digunakan dikarenakan dengan menerapkan *sharpening* pada citra, citra *output* yang dihasilkan menjadi lebih mudah untuk diinterpretasikan (Priyawati, 2011).

Pada operasi penajaman, operasi ini bekerja dengan menekan frekuensi rendah pada citra dan memperkuat elemen yang frekuensinya tinggi seperti bagian tepi atau perbatasan objek pada citra sehingga memberi dampak pada pinggiran objek yang terlihat lebih kuat/jelas dibanding sekelilingnya. Karena inilah operasi penajaman citra atau *image sharpening* lebih dikenal dengan teknik mempertajam tepi (*edge sharpening*) atau peningkatan kualitas tepi (*edge enhancement*) (Munir, 2004). Contoh operasi ini adalah *high-pass filter* atau penapis lolos tinggi.



Gambar 2.8 (a) Citra Lena Awal dan (b) Citra Lena Hasil *Image Sharpening* (Munir, 2004)

2.8. *High-Pass Filter* (HPF)

High-Pass Filter (HPF) atau penapis lolos tinggi adalah salah satu dari metode penajaman (*sharpening*). Dimana tujuan utama dari proses penajaman adalah untuk menyoroti detail-detail halus dalam gambar atau untuk meningkatkan detail yang telah dikaburkan baik dalam kesalahan atau efek alami dari proses akuisisi citra tertentu. *High-pass filter* merupakan contoh dari filter yang di

terapkan dalam menapis sinyal berfrekuensi tinggi (misal seperti titik halus, detail halus, tepi dan juga garis halus), dengan kata lain mengambil informasi citra pada frekuensi tinggi dan melepaskan data rendah frekuensinya. Oleh karena itu, tapis *high-pass* sering disebut juga sebagai filter penajaman tepi (*edge sharpening*) karena HPF digunakan dalam proses penajaman tepi-tepi objek yang sering digunakan dalam aplikasi deteksi tepi (*edge detection*) objek.

Tapis lolos tinggi (*high-pass filter*) merupakan salah satu metode berdomain spasial. Teknik pemrosesan sebuah citra yang berdasarkan metode spasial beroperasi langsung pada pikselnya. Metode-metode ini memodifikasi nilai-nilai piksel sesuai dengan aturan yang tergantung pada nilai piksel asli yaitu proses lokal atau titik, sehingga membuat domain berbasis spasial mudah dipahami dan berkurangnya kompleksitas yang membuatnya sederhana secara konseptual. Adapun aturan-aturan dalam *high-pass filter* (tapis lolos tinggi) (Munir, 2004)

1. Koefisien penapis boleh negatif, nol, ataupun bernilai positif.
2. Total keseluruhan koefisiennya ialah bernilai 0 ataupun 1.

Apabila jumlah koefisiennya berjumlah = 0, maka setiap elemen yang rendah frekuensinya nilainya akan menurun. Namun, apabila total dari koefisien adalah = 1, maka elemen yang memiliki frekuensi rendah nilainya tetap sama dengan nilai semula.

$$\sum_{xy} H(x, y) = 0 \dots\dots\dots (2.9)$$

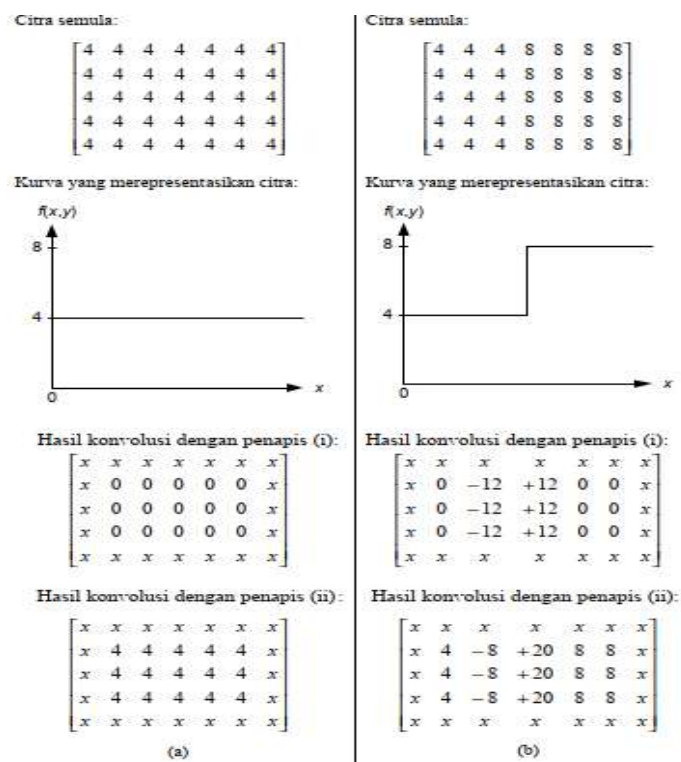
Adapun contoh-contoh kernel pada pada tapis lolos tinggi (*high-pass filter*) adalah sebagai berikut:

$$\begin{array}{ll} \text{(i) } \Sigma = 1 \begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix} & \text{(ii) } \Sigma = 0 \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} \\ \text{(iii) } \Sigma = 0 \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} & \text{(iv) } \Sigma = 1 \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix} \end{array}$$

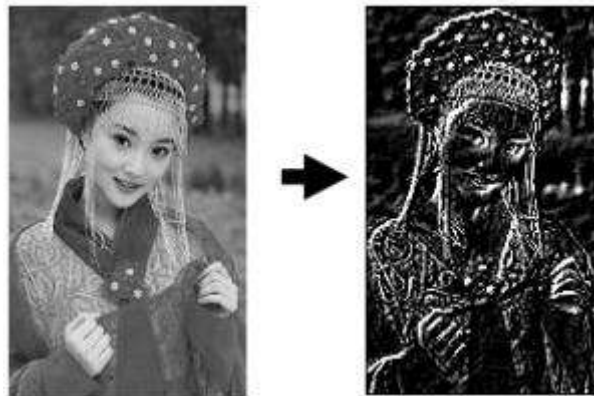
$$(v) \Sigma = 0 \begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix}$$

$$(vi) \Sigma = 1 \begin{bmatrix} 1 & -2 & 1 \\ -2 & 5 & -2 \\ 1 & -2 & 1 \end{bmatrix}$$

Proses konvolusi pada *high-pass filter* sangat berpengaruh pada nilai koefisien yang banyak di titik tengah penapis yang mana mempunyai peranan yang sangat penting. Pada sebuah citra yang memiliki frekuensi yang tinggi, nilai titik pusat/tengah dikalikan dengan piksel yang dihitung. Koefisien yang nilainya negatif akan lebih kecil dari sekitar titik tengah penapis bekerja untuk mengecilkan faktor pembobotan yang banyak. Dimana efek *nettonya* adalah, setiap piksel diperkuat jika piksel bernilai besar, sedangkan pada area citra yang mempunyai intensitas piksel yang tetap/stabil tidak berubah nilainya.



Gambar 2.9 Contoh Hasil Konvolusi Pada Tapis HPF: (a) Citra Mempunyai Piksel Tepi, (b) Citra Tidak Ada Piksel Tepi (Munir, 2004)



Gambar 2.10 Citra HPF Dengan Nilai Koefisien = 0 (Imron, 2013)

2.9. Fast Fourier Transform (FFT)

Ditahun 1965 ditemukanlah bentuk transformasi fourier yang cukup populer di bidang pengolahan citra yaitu dikenal *Fast fourier transform* (FFT). Jean Baptiste Joseph Fourier di tahun 1822 sebagai penemu bentuk transformasi fourier sebelumnya dan dia berhasil mengembangkannya sehingga menjadi *Fast Fourier Transform*. Transformasi fourier adalah sebuah metode yang di terapkan untuk mengganti gelombang seismik dari ranah waktu (*time domain*) diubah ke ranah frekuensi (*frequency domain*). Adapun fungsi dilakukannya transformasi ini agar memperoleh informasi apakah suatu sinyal ada frekuensinya atau tidak ada. Transformasi fourier membagi sebuah sinyal menjadi kedalam frekuensi saling berlainan di fungsi eksponensial yang kompleks (Sipasulta et al., 2014).

Transformasi fourier bekerja dengan mengubah fungsi dari ranah spasial ke berganti ke ranah frekuensi, sedangkan untuk melakukan perubahan terhadap transformasi fourier dilakukan dengan menerapkan transformasi fourier balikkannya (*invers*). Transformasi fourier 2D yang berasal dari suatu fungsi $f(x, y)$ dapat didefenisikan dengan persamaan berikut:

$$F(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-j2\pi(ux/M + vy/N)} \dots\dots\dots (2.10)$$

Dimana $u = 0, 1, 2 \dots M-1$; $v = 0, 1, 2 \dots N-1$. Kemudian *invers* dari transformasi fourier diatas adalah:

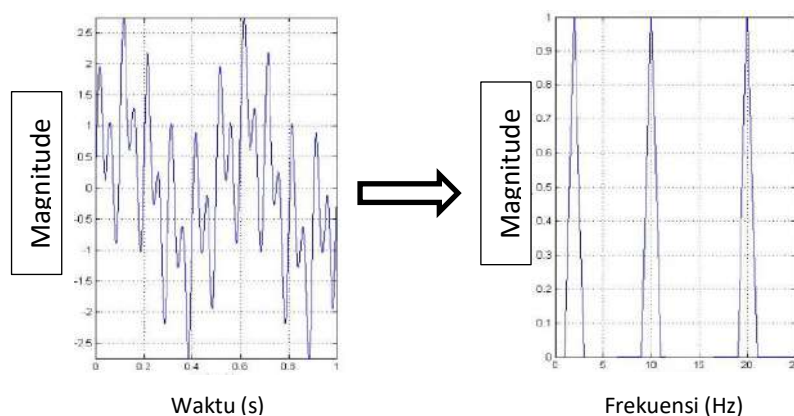
$$f(x, y) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) e^{j2\pi (ux/M + vy/N)} \dots\dots\dots (2.11)$$

Dimana $x = 0, 1, 2 \dots M-1$; $y = 0, 1, 2 \dots N-1$. Pada proses *invers* citra asli akan di transformasi dengan transformasi fourier untuk bisa di ubah dari domain spasial ke domain frekuensi, sehingga hasilnya akan dikalikan dengan *filtering*. Adapun sifat-sifat dari transformasi fourier antara lain:

Tabel 2.2 Sifat-Sifat Transformasi Fourier (Munir, 2004)

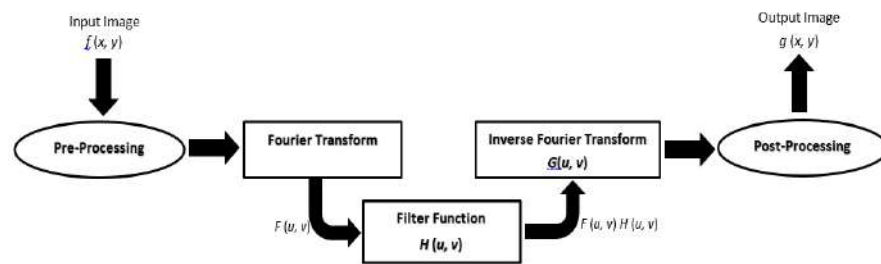
No.	Sifat	Ranah Waktu	Ranah Frekuensi
1	Kelanjutan	$af(t) + bg(t)$	$aF(u) + bG(u)$
2	Penskalaan	$f(at)$	$\frac{1}{ a } F(u/a)$
3	Pergeseran	$f(t - a)$	$F(u - a)$
4	Modulasi	$e^{i2\pi at} f(t)$	$f(u) e^{-i2\pi ua}$
5	Konyugasi	$f^*(t)$	$F^*(-u)$
6	Konvolusi	$h(t) = f(t) * g(t)$	$H(u) = F(u)G(u)$
7	Perkalian	$h(t) = f(t)g(t)$	$H(u) = F(u) * G(u)$
8	Diferensial	$\frac{d^n f(t)}{dt^n}$	$(i2\pi u)^n F(u)$
9	Simetri	$F(t)$	$f(-u)$
10	Hasil Kali Dalam	$\int_{-\infty}^{\infty} f(t)g^*(t)dt$	$\int_{-\infty}^{\infty} F(u)G^*(u)du$

Dibawah ini merupakan contoh transformasi fourier dari domain waktu ke domain frekuensi:



Gambar 2.11 Transformasi Fourier dari Domain Waktu ke Domain Frekuensi
(Chahyati, 2004)

Fast fourier transform sebagai sebuah algoritma untuk menghitung transformasi fourier diskrit (DFT) dengan cepat dan juga efisien. Dalam penerapannya, *fast fourier transform* (FFT) digunakan dalam berbagai bidang, seperti memecahkan persamaan diferensial parsial, pengolahan sinyal digital, dan sebagai algoritma dalam mengalikan bilangan bulat besar.

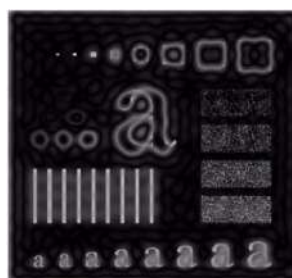


Gambar 2.12 Langkah Dasar *Filtering* Dalam Domain Frekuensi

Pada pengolahan citra digital, *fast fourier transform* bisa diterapkan dalam mempertajam citra (*image sharpening*), dimana pada domain frekuensi ada tiga metode *filter* yang digunakan, yaitu:

1. *Ideal High-Pass Filter* (IHPF)

$$H(u, v) = \begin{cases} 0 & \text{if } D(u, v) \leq D_0 \\ 1 & \text{if } D(u, v) > D_0 \end{cases} \dots\dots\dots (2.12)$$



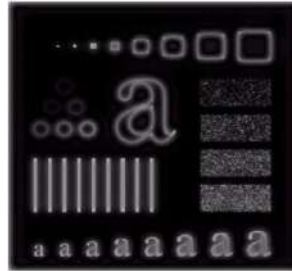
Gambar 2.13 IHPF Dengan $D_0=15$ (Ferrarese, 2002)

Dimana D_0 adalah *Cut-Off* frekuensi, $D(u, v)$ adalah jarak dari titik (u, v) ke pusat dari kotak frekuensi.

2. *Butterworth High-Pass Filter* (BHPF)

Hasil dari *Butterworth High-Pass Filter* (BHPF) order 2 pada D_0 yang diset sama dengan IHPF, tampilannya akan tampak lebih lembut (*smooth*) dari hasil penerapan IHPF (Siregar & Aryanta, 2013).

$$H(u, v) = \frac{1}{1 + [D_0 / D(u, v)]^{2n}} \dots\dots\dots (2.13)$$

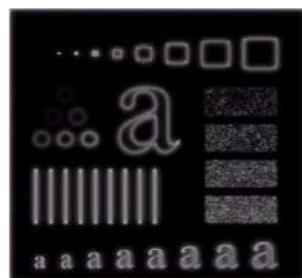


Gambar 2.14 BHPF Dengan Order 2 $D_0 = 15$ (Ferrarese, 2002)

3. *Gaussian High-Pass Filter* (GHPF)

Fungsi transfer pada *Gaussian High-Pass Filter* (GHPF) dengan *cutoff* frekuensi terletak pada jarak D_0 dari origin, yaitu:

$$H(u, v) = 1 - e^{-D^2(u, v) / 2 D_0^2} \dots\dots\dots (2.14)$$



Gambar 2.15 GHPF Dengan $D_0 = 15$ (Ferrarese, 2002)

2.10. Histogram Ekualisasi (*Histogram Equalization*)

Histogram ekualisasi merupakan suatu teknik untuk mengatur intensitas citra agar kualitasnya lebih baik (Sri Arsa et al., 2017). Proses histogram ekualisasi sebagai suatu cara mengarah agar diperolehnya histogram citra dimana nilai derajat keabuannya terdistribusi secara seragam pada sebuah citra (Rahmawati et al.,

2018). Umumnya metode ini di pakai untuk mengembangkan jarak (*range*) level nilai keabuan sehingga memberikan dampak meningkatkan kontras citra itu. Teknik ini sangat efektif dalam meningkatkan tingkat kecerahan maupun ketajaman seluruh citra serta dapat meningkatkan detail tekstur. Dengan menurunkan rumus perataan histogram, secara matematis dapat dijabarkan sebagai berikut :

$$S_k = T(r_k) = (L - 1) \sum_{j=0}^k P_r(r_j) = \frac{L-1}{MN} \sum_{j=0}^k n_j \quad \dots\dots\dots (2.15)$$

Dimana $k = 0, 1, 2, 3, \dots, L-1$.

Dalam persamaan transformasi histogram ekualisasi pada citra digital, variabel $M \times N$ menghasilkan total dari piksel, L jumlah tingkat keabuan, dan $p_r(r_j)$ jumlah piksel dalam gambar masukan dengan intensitas nilai r_j . Rentang dari nilai *input* serta *output* dari level keabuan berada dikisaran $0, 1, 2, \dots, L-1$. Kemudian transformasi dari histogram ekualisasi memetakan input nilai r_k (dimana $k = 0, 1, 2, \dots, L-1$) sampai nilai *output* S_k .



Gambar 2.16 Citra Asli (a) dan Citra Hasil Histogram Ekualisasi (b) (Arif et al., 2018)

2.11. Pemotongan (*Cropping*)

Pada pengolahan citra digital, dimana dalam prosesnya hanya membutuhkan bagian atau daerah tertentu dari citra yang akan diolah. Bagian atau daerah yang diinginkan tersebut dikenal dengan istilah *Region of Interest* (ROI).

Salah satu proses agar mendapatkan citra ROI (*Region of Interest*) adalah dengan menggunakan proses *cropping*. Penerapan ini agar hanya memakai citra yang akan diolah dan membuang citra yang tidak diperlukan. Posisi citra yang akan di *cropping* di sesuaikan menurut citra itu sendiri ini di sebabkan ukuran telapak tangan setiap individu tidaklah sama. Proses *cropping* ini diterapkan dengan menentukan koordinat-koordinat pada citra. Dalam penelitian ini, citra di *crop* sebelum di *input* ke sistem untuk di ambil bagian tertentu saja yaitu bagian dalam telapak tangan saja (tidak termasuk jari-jari).

2.12. *Resize*

Resize adalah suatu proses untuk mengubah *size* piksel citra, baik itu menjadi lebih besar atau bisa saja lebih kecil dari citra sebelumnya. Proses ini diperlukan karena dapat mempermudah pemrosesan citra ketahap berikutnya, dikarenakan beberapa citra *input* yang akan dipakai mempunyai ukuran resolusi yang besar sehingga dapat memperlambat kecepatan sistem dan hasil pemrosesan yang kurang baik. Dalam melakukan *resize* pada citra, proses penurunan resolusi pada citra dilakukan dengan membandingkan antara jumlah baris serta kolom. Dalam kasus penelitian ini, gambar telapak tangan yang diedit akan diubah ukurannya menjadi 256x256 piksel.

2.13. PSNR Dan MSE

Dalam memperkirakan kualitas suatu gambar, diperlukan uji target dengan memastikan nilai MSE dan PSNR dari gambar hasil. Penggunaan MSE dan PSNR sebagai parameter dipakai untuk indikator dalam membandingkan tingkat kesamaan citra hasil. MSE dan PSNR digunakan untuk melihat konsekuensi dari gambar asli dengan gambar hasil.

Mean Square Error (MSE) adalah kesalahan harga yang dikuadratkan dengan kesalahan normal antara gambar pertama dan gambar dari hasil olah. Secara matematis dapat dituliskan persamaan sebagai berikut:

$$MSE = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} |(f(x, y) - g(x, y))|^2 \quad \dots \dots \dots (2.16)$$

Semakin rendah nilai dari *Mean Square Error* (MSE) citra hasil maka proses rekonstruksi sangat baik. *Peak Signal to Noise Ratio* (PSNR) merupakan parameter yang sering digunakan untuk proses kompresi sebuah citra dalam menentukan kualitas hasil rekonstruksi citra *output*/hasil. Setelah memperoleh hasil MSE, nilai PSNR ditentukan dari kuadrat nilai paling tinggi yang dibagi oleh hasil MSE.. Berikut ini adalah persamaan matematis dari *Peak Signal to Noise Ratio* (PSNR):

$$PSNR = 10 \cdot \log_{10} \left(\frac{MAX^2}{MSE} \right) \dots\dots\dots (2.17)$$


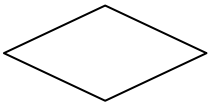
Satuan dari *Peak Signal to Noise Ratio* (PSNR) adalah desibel (dB). Semakin besar nilai estimasi PSNR tersebut, semakin baik pula sifat gambar hasil yang dibuat (Palanikumar et al., 2011), dimana dua gambar dengan derajat kedekatan rendah jika nilai PSNR <30 dB.




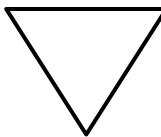


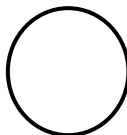

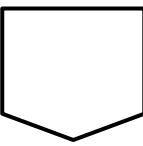

2.14. *Flowchart*

Diagram alir (*flowchart*) adalah diagram dengan gambar eksplisit yang memiliki alur yang menggambarkan pengaturan langkah-langkah pengaturan dalam suatu masalah. Diagram alir adalah suatu pendekatan untuk memperkenalkan suatu algoritma. Alasan pembuatan diagram alur adalah sebagai berikut:

1. Menggambarkan suatu tahapan penyelesaian masalah
2. Secara sederhana, terurai, rapi dan jelas
3. Menggunakan simbol-simbol standar

Tabel 2.3 Simbol *Flowchart*

	<i>Flow Direction</i>		<i>Decision</i>
	Arah aliran program yang menghubungkan simbol lainnya.		Penentuan siklus bergantung pada kondisi eksisting.

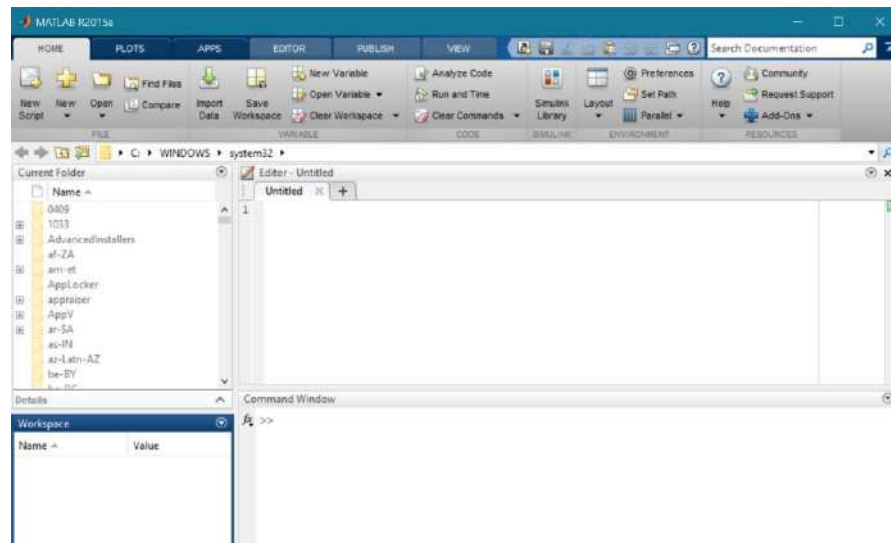
	<i>Terminator</i> Awal/akhir program		<i>Predefine Process</i> Gambar untuk pelaksanaan bagian (subbagian) / teknik.
	<i>Preparation</i> Proses inisialisasi/ pemberian harga		<i>Offline Storage</i> Menunjukkan bahwa informasi pada gambar ini akan disimpan ke media yang telah ditentukan
	<i>Process</i> Proses pengolahan data yang dilakukan oleh komputer		<i>Document</i> Dokumen atau laporan
	<i>On Page Connector</i> Antarmuka bagian diagram alir di halaman yang setara.		<i>Input/Output Data</i> Interaksi input / hasil tidak terlalu memperhatikan jenis peralatan
	<i>Off Page Connector</i> Area diagram alur koneksi di berbagai halaman		<i>Manual</i> Mengatasi siklus yang tidak dilakukan di PC

2.15. Matlab (*Matrix Laboratory*)

Matrix Laboratory atau dikenal dengan sebutan Matlab, merupakan bahasa pemrograman dengan berbagai kapasitas dan kualitas dari bahasa pemrograman seperti Delphi, C ++ atau Basic. Sebagai bahasa pemrograman tingkat signifikan yang berkomitmen untuk kebutuhan komputasi teknis, visualisasi dan pemrograman seperti komputasi matematik, pengembangan algoritma, analisis data, simulasi dan pemodelan dan grafik-grafik perhitungan.

Berikut ini adalah beberapa karakteristik dari pemrograman Matlab (Ariyanto & Caesarendra, 2011):

1. Bahasa pemrograman bergantung pada matriks (baris dan kolom)
2. Ada banyak simpanan alat untuk aplikasi, misalnya; *Simulink*, *Neural Network*, *State Flow*, *Data Acquisition Toolbox*, *Communications Blockset*, *Fuzzy Logic Toolbox*, *Image Acquisition Toolbox*, *Signal Processing Blockset*, dan lain sebagainya.
3. Dalam menulis kode programnya, tidak harus mendeklarasikan *array* terlebih dahulu.
4. Memiliki waktu perbaikan program yang lebih cepat dibandingkan dengan pemrograman konvensional, misalnya Fortran, dan C.



Gambar 2.17 Tampilan Matlab Versi R2015a

Beberapa bagian utama dari *window* Matlab adalah sebagai berikut:

1. *Command Window*

Ini adalah jendela utama Matlab. Dimana pada bagian ini terdapat tempat untuk menjalankan kapasitas, menjalankan ukuran, menyebutkan faktor, dan melihat variabel substansi.

2. *Current Directory*

Jendela ini menunjukkan substansi dari registri yang berfungsi saat menggunakan Matlab. Selanjutnya, indeks dapat diubah dengan direktori kerja yang ideal.

3. *Workspace*

Berfungsi untuk menampilkan semua faktor yang sudah aktif saat menggunakan Matlab. Jika variabel adalah matriks yang sangat besar, klien dapat melihat substansi semua informasi dengan mengetuk dua kali pada variabel. Matlab secara alami akan menampilkan jendela "*array editor*" yang berisi informasi untuk setiap faktor yang dipilih oleh klien.

4. *Command History*

Jendela ini digunakan untuk menyimpan pesan yang baru-baru ini dilakukan oleh klien terhadap Matlab.

2.16. Penelitian Relevan

Tabel 2.4 Penelitian Relevan

No.	Peneliti	Asal & Tahun	Judul	Kesimpulan
1	Salahuddin, Tulus, Fahmi F	Politeknik Negeri Lhokseumawe dan Universitas Sumatera Utara, Indonesia (2013)	<i>Klasifikasi dan Peningkatan Kualitas Citra Sidik Jari Menggunakan FFT (Fast Fourier Transform)</i>	Peningkatan kualitas gambar kesan jari kering dengan teknik FFT mendapatkan nilai $K = 0,6$ yang optimal. Dengan persentase <i>enhancement</i> sidik jari $K = 0,6$ sebesar 97,52% dan <i>matching</i> sidik jari tertinggi adalah $K = 0,6$ sebesar 54,29%. Kemudian hasil dari sidik jari menunjukkan bahwa semua informasi sidik jari dinyatakan kering karena

				kemungkinan jenis kulit yang diambil kering dan ukuran variasi informasi tidak terlalu bagus. (Fahmi, 2013)
2	S. Palanikumar, M. Sasikumar, J. Rajeesh	Noorul Islam College Of Engineering, India (2011)	<i>Palmprint Enhancement Using Discrete Curvelet Transform</i>	Setelah mengevaluasi dari 100 tes yang dilakukan, hasil penelitian menunjukkan bahwa metode yang diusulkan yaitu <i>Discrete Curvelet Transform</i> (DCT) memberikan hasil yang lebih baik. Strategi ini dapat menghilangkan derau dalam gambar yang dikontraskan dengan metode <i>adaptive histogram equalization</i> (AHE), <i>histogram equalization</i> (HE) dan <i>unsharp masking</i> (UM) sehingga lebih banyak ciri yang dapat diekstrak dari telapak tangan (Palanikumar et al., 2011).
3	Eka Rahmawati, Ragil Wijanto, Dwiza Riana	STMIK Nusa Mandiri, Indonesia (2018)	<i>Perbandingan Metode Peningkatan Kualitas Citra dan Deteksi Tepi pada</i>	Pemakaian operasi titik, <i>fast fourier transform</i> , <i>neighborhood averaging</i> , <i>histogram equalization</i> , dan <i>median filtering</i> pada tahap <i>pre-processing</i> menunjukkan fakta bahwa operasi deteksi

			<i>Citra Kutu Kebul</i>	tepi <i>sobel</i> , <i>isotropic</i> , dan <i>gradient</i> memberikan citra kutu kebul yang lebih baik daripada dengan operasi <i>canny</i> yang mana lebih menonjolkan pola dari daun (Rahmawati et al., 2018).
4	Agustinus Siregar, Dwi Aryanta	Institut Teknologi Nasional Bandung, Indonesia (2013)	<i>Simulasi Dan Analisis Perbaikan Citra Digital Domain Frekuensi Dengan Transformasi Fourier</i>	Pemakaian metode <i>Gaussian Low-Pass Filter</i> sebagai metode yang baik dalam proses penghalusan citra terhadap citra sinar X dan pada proses penajaman citra terhadap citra sinar X metode yang baik yakni <i>Ideal High-Pass Filter</i> (IHPF) (Siregar & Aryanta, 2013).
5	Diah Priyawati	Universitas Muhammadiyah Surakarta, Indonesia (2011)	<i>Teknik Pengolahan Citra Digital Berdomain Spasial Untuk Peningkatan Citra Sinar-X</i>	Proses <i>image enhancement</i> membuktikan dapat mengoptimalkan kualitas citra sinar-x yang mengalami penurunan kualitasnya saat digitalisasi citra. Beberapa teknik yang memiliki pengaruh yang besar seperti teknik histogram ekualisasi dan penajaman citra (<i>image sharpening</i>) (Priyawati, 2011).

BAB III

METODOLOGI PENELITIAN

3.1. Lokasi Dan Waktu Penelitian

3.1.1. Lokasi Penelitian

Adapun lokasi penelitian ini dilakukan, yaitu berada di Laboratorium Multimedia Fakultas Sains dan Teknologi yang berada di Jalan IAIN No. 1 Medan, Sumatera Utara.

3.1.2. Waktu Penelitian

Adapun waktu penelitian ini dilakukan yaitu dari Maret 2020 sampai dengan September 2020, yang dapat dilihat dari tabel berikut ini:

Tabel 3.1 Jadwal Dan Waktu Penelitian

No.	Rencana Kegiatan	Jadwal Kegiatan						
		Maret 2020	April 2020	Mei 2020	Juni 2020	Juli 2020	Agustus 2020	September 2020
1.	Pra-Penelitian							
	<i>a. Penyusunan dan Pengajuan Judul</i>							
	<i>b. Pengajuan Proposal</i>							
2.	Pelaksanaan							
	<i>a. Pengumpulan Data</i>							
	<i>b. Analisis Data</i>							
3	Tahap Penyusunan Laporan							

3.2. Bahan Dan Alat Penelitian

3.2.1. Bahan

Adapun penelitian ini dilakukan dengan menggunakan citra telapak tangan khususnya bagian dalam (tidak termasuk jari) sebagai bahan penelitian demi mendukung serta menunjang pelaksanaan penelitian.

3.2.2. Perangkat Keras (*Hardware*)

Adapun perangkat keras (*hardware*) yang digunakan dalam proses pembuatan sistem ini adalah sebagai berikut:

1. *System Model* : Latitude E7440
2. *Processor* : Intel® Core™ i3 4030U CPU @ 1.90GHz (4 CPUs)
3. *Memori* : *Random Acces Memory* (RAM) 4096 MB
4. *Penyimpanan* : *Solid State Drive* (SSD) 128 GB

3.2.3. Perangkat Lunak (*Software*)

Adapun perangkat lunak (*software*) yang digunakan dalam proses penelitian ini adalah sebagai berikut:

1. *Operating System Windows* 10 Pro 64-bit
2. *Matlab* R2015a
3. *Microsoft Office Word* 2013
4. *Microsoft Visio Pro* 2016

3.3. Prosedur Kerja

3.3.1. Perencanaan

Adapun rencana kegiatan yang dilakukan dalam penelitian ini, dengan melalui beberapa tahap yang dapat dilihat sebagai berikut:



Gambar: 3.1. Tahapan Prosedur Kerja

3.3.2. Teknik Pengumpulan Data

Dalam proses pengumpulan data ada beberapa metode yang digunakan. Strategi yang digunakan dalam mengumpulkan informasi di pengujian ini adalah teknik penelitian pustaka dan prosedur *sampling*. Berikut akan dijelaskan teknik-teknik yang digunakan dalam interaksi penelitian ini.

1. Penelitian Kepustakaan (*Library Research*)

Penelitian kepustakaan pada penelitian ini dilakukan dengan cara mengumpulkan informasi terkait dengan penelitian, yang nantinya akan digunakan sebagai acuan dalam proses pembuatan sistem dan penyusunan tugas akhir ini. Data ataupun informasi yang didapat bisa berasal dari buku-buku ilmiah, jurnal penelitian terkait, artikel, laporan penelitian, skripsi serta sumber referensi tertulis lainnya yang berkaitan dengan penelitian ini seperti teori-teori tentang *image processing*, biometrika telapak tangan, *image enhancement* dan metode-metodenya, serta program Matlabnya.

2. Teknik *Sampling*

Proses pengambilan sampel di penelitian ini, yaitu dengan menggunakan *purposive random sampling* (pengambilan menurut tujuan). Tes yang diambil dalam bermacam-macam informasi dalam pengujian ini dilengkapi dengan pengumpulan gambar telapak tangan. Citra telapak tangan yang akan digunakan sebanyak 30 citra, yang di dapat penulis dari basis data telapak tangan (*palmprint database*). Kriteria citra telapak tangan yang digunakan harus jelas (terkhusus tampak bagian dalam telapak tangan), dan kondisi telapak tangan normal (tidak berminyak, kotor ataupun ada bekas luka).

3.3.3. Analisis Kebutuhan

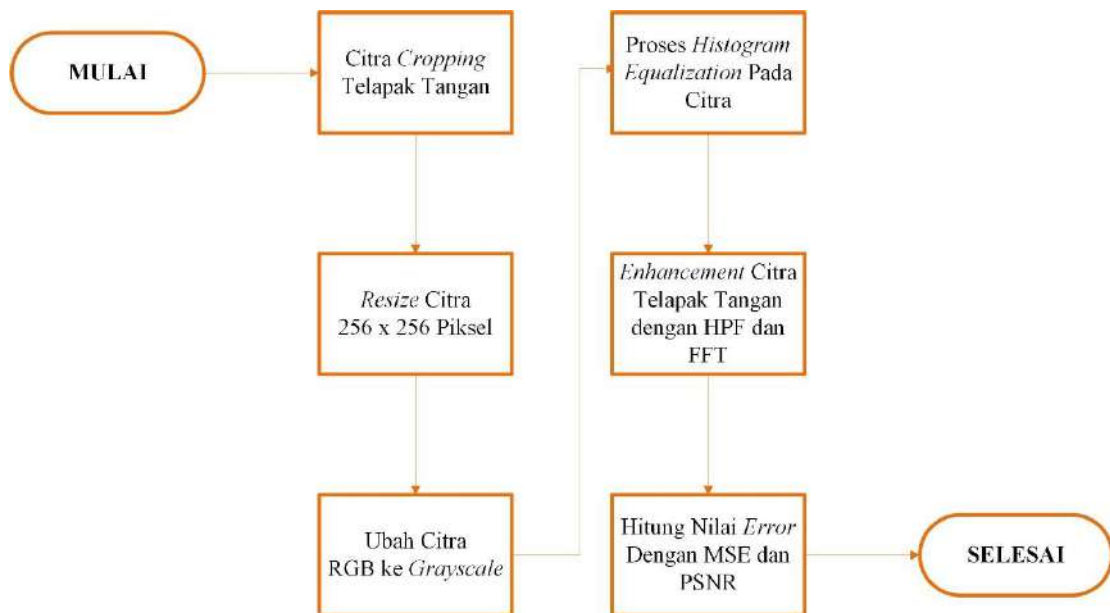
Analisis kebutuhan merupakan suatu proses agar mendapatkan informasi-informasi tentang apa saja kebutuhan dalam proses rekayasa sistem yang nantinya akan digunakan dalam membangun sistem yang sesuai dengan kebutuhan tersebut. Alasan dilakukannya analisis kebutuhan yaitu agar batasan dari sistem yang akan dirancang, menentukan kemampuan serta fungsi sistem sesuai dengan kebutuhan yang ada.

Citra telapak tangan merupakan salah satu kebutuhan pada sistem ini yang akan digunakan sebagai masukan (*input*). Pada citra telapak tangan, terdapat detail-detail yang sering digunakan dalam penelitian sistem biometrik. Namun, hasil citra tersebut terkadang ada gangguan dalam proses identifikasi. Bisa dikarenakan pada citra nya sendiri atau ciri yang terdapat pada telapak tangan yang kurang baik/jelas.

Proses *image enhancement* merupakan salah satu tahap dari pengolahan citra yang sangat dibutuhkan dalam memperbaiki kualitas pada citra. Banyak sekali metode-metode yang digunakan dalam *image enhancement*. Penggunaan metode tersebut dapat menyelesaikan permasalahan-permasalahan yang ada pada citra yang akan digunakan. Dalam penelitian ini, metode yang digunakan adalah *High-Pass Filter* (HPF) dan *Fast Fourier Transform* (FFT). Penggunaan kedua metode ini di terapkan pada proses *image sharpening* (penajaman citra). Pada kasus penelitian kali ini, citra telapak tangan yang akan di olah akan di pertajam citranya sehingga dapat memperjelas ciri yang ada pada telapak tangan.

3.3.4. Perancangan Sistem

Tahap perancangan merupakan tahap selanjutnya setelah analisis kebutuhan. Perancangan merupakan tahap konseptualisasi dimana pada tahap ini diharuskan analisis dalam perancangan sistemnya agar tahu tentang apa saja yang menjadi kebutuhan dan keinginan pengguna (*user*), sehingga sistem yang akan di buat dapat sesuai dengan kebutuhan dalam mengatasi permasalahan yang ada. Kebutuhan tersebut bisa berupa peralatan *hardware* atau *software*, maupun gambaran sistem secara detail. Berikut ini akan disajikan diagram dan *flowchart* dalam penelitian ini:

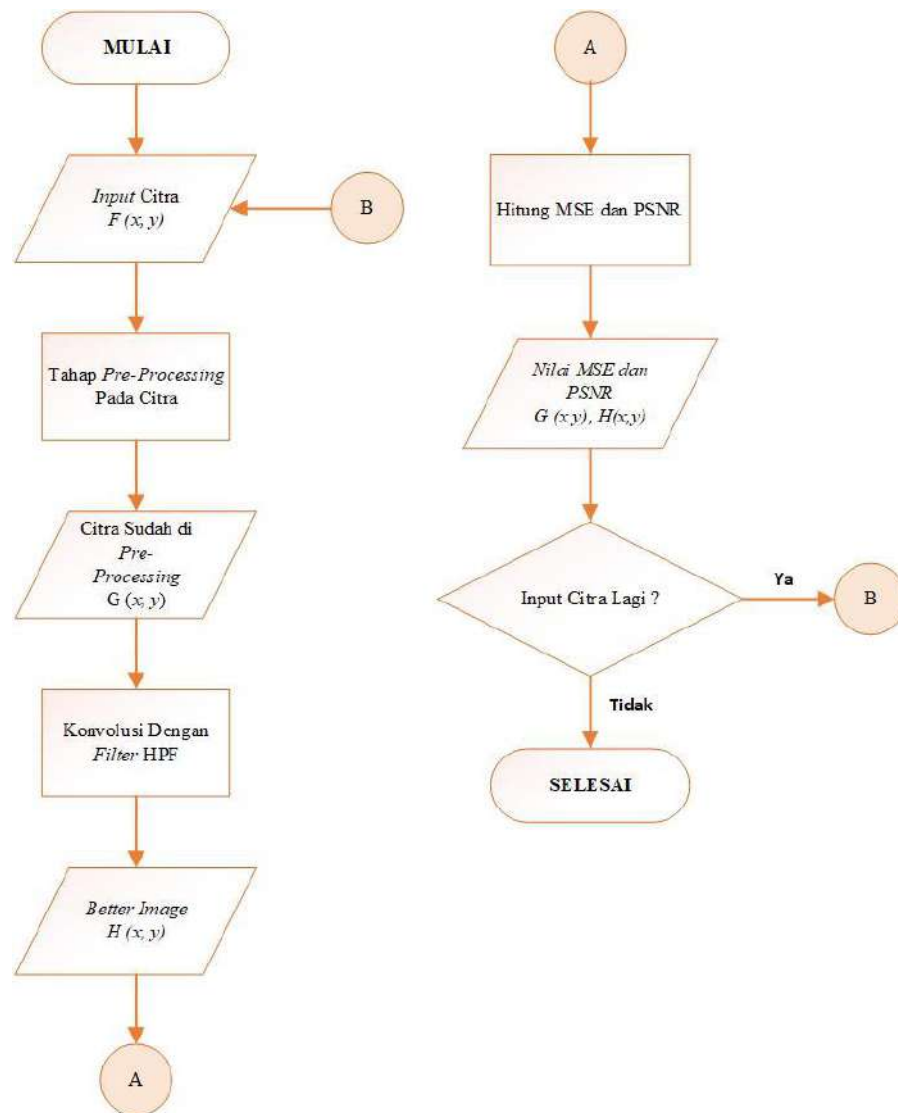


Gambar: 3.2. Diagram Blok Perencanaan Sistem

Dapat dilihat pada gambar diagram perencanaan diatas yang menjelaskan alur dari suatu sistem dalam melakukan proses *enhancement* pada sebuah citra telapak tangan. Citra yang di *input* merupakan citra telapak tangan yang sebelumnya sudah dilakukan proses *cropping* terlebih dahulu. Kemudian, citra di *input* kedalam sistem untuk dilakukan proses *pre-processing*. Proses *resize* pada citra dilakukan agar semua ukuran citra menjadi sama rata yang nantinya akan mempermudah/mempercepat pemrosesan citra ketahap berikutnya, dalam penelitian ini citra di *resize* menjadi ukuran 256 x 256 piksel. Selanjutnya konversi

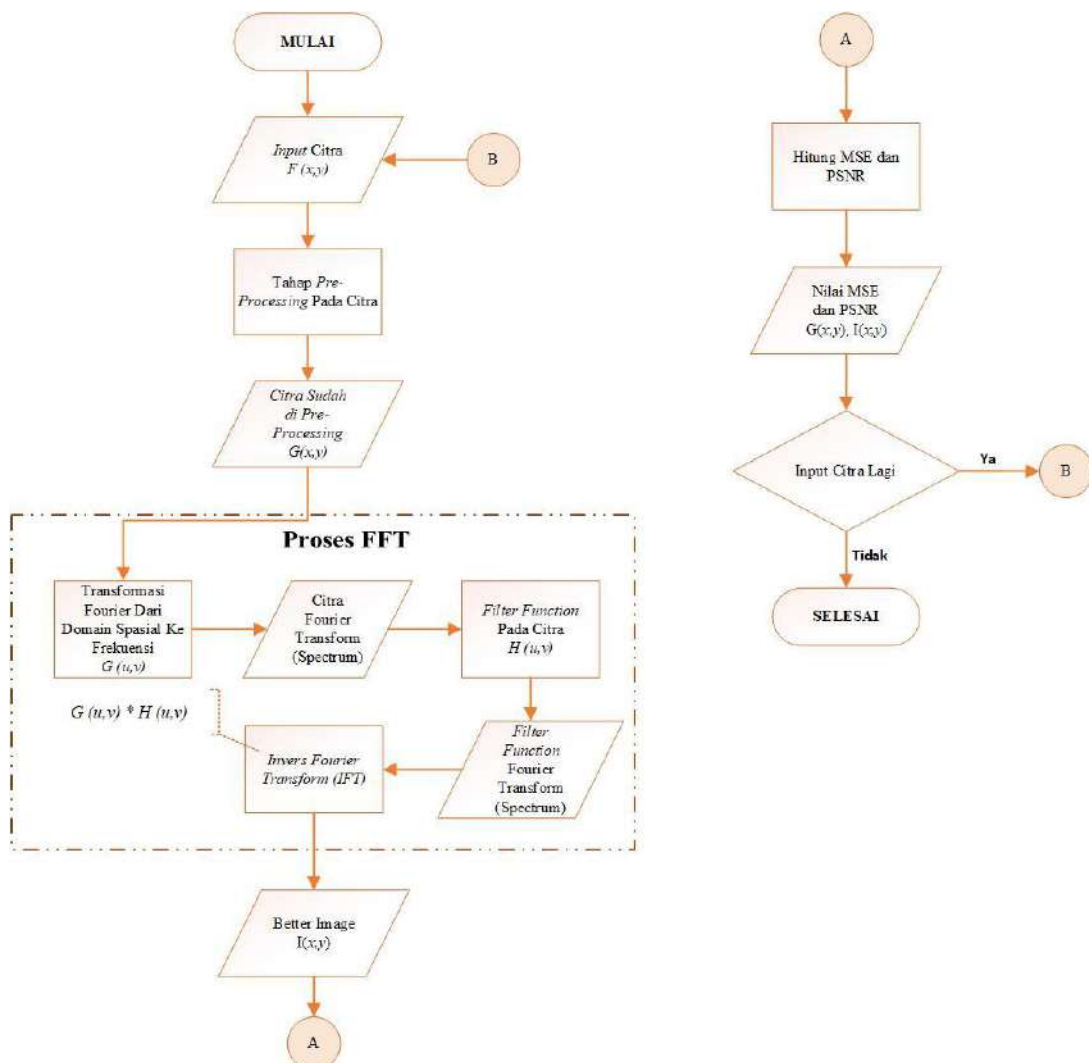
dari citra RGB di ubah ke bentuk *grayscale*. Tahap *pre-processing* berikutnya yaitu dengan menerapkan histogram ekualisasi. Terkadang intensitas/kontras pada citra tidak sama dengan yang lain, hal ini bisa terjadi saat proses akuisisi citra. Histogram ekualisasi diterapkan pada citra telapak tangan untuk memperbaiki kecerahan dengan meningkatkan kontras/intensitas pada citra tersebut.

Setelah dilakukan tahap *pre-processing*, maka selanjutnya citra tersebut di proses dengan menerapkan metode *high-pass filter* (HPF) dan *fast fourier transform* (FFT). Dalam penerapan metode HPF pada citra, HPF akan bekerja secara langsung pada piksel-piksel citra, dengan memodifikasi nilai piksel asli menggunakan kernel HPF. Sedangkan pada *fast fourier transform* (FFT), citra terlebih dahulu diubah dari domain spasial ke domain frekuensi. Setelah citra dalam bentuk domain frekuensi, maka dilakukanlah fungsi *filter* pada citra. Pada domain frekuensi, ada 3 fungsi filter seperti *ideal high-pass filter* (IHPF), *butterworth high-pass filter* (BHPF), dan *gaussian high-pass filter* (GHPF). Selanjutnya untuk mendapatkan citra yang sudah di *enhancement*, citra harus dikembalikan lagi ke domain spasial dengan menerapkan *invers* pada citra. Setelah kedua metode selesai dan citra yang sudah di *enhancement* didapatkan, maka dihitunglah nilai *error* citra dengan menggunakan MSE dan PSNR. Nilai *error* dihitung untuk memahami kualitas pada citra yang sudah di *enhancement* dan setelah nilai *error* didapatkan maka proses *enhancement* pada citra telapak tangan selesai.



Gambar: 3.3. Flowchart High-Pass Filter (HPF)

Adapun keterangan dari gambar *flowchart* diatas adalah, setelah citra yang berformat BMP di *input*, maka dilakukan *pre-processing* pada citra. Proses *pre-processing* ini meliputi mengubah citra ke *grayscale*, melakukan *resize* pada citra dan meningkatkan kontras citra dengan melakukan histogram ekualisasi. Maka selanjutnya adalah dengan melakukan filter *high-pass filter* (HPF). Setelah didapatkan citra yang sudah di *enhancement*, maka dihitunglah nilai *error* pada citra dengan menggunakan MSE dan PSNR.



Gambar: 3.4. Flowchart Fast Fourier Transform (FFT)

Sama halnya dengan flowchart HPF, setelah dilakukan proses *pre-processing* pada citra telapak tangan. Maka selanjutnya adalah dengan menerapkan metode *fast fourier transform* (FFT). Dalam metode ini, dilakukanlah fungsi *filter* pada citra yang sudah dalam bentuk domain frekuensi, kemudian dikembalikan lagi ke domain spasial untuk mendapatkan citranya kembali atau disebut dengan istilah *invers*. Setelah didapatkan citra yang sudah di *enhancement*, maka dihitunglah nilai *error* pada citra dengan menggunakan MSE dan PSNR.

3.3.5. Pengujian

Tahap pengujian kerangka kerja adalah cara untuk menguji produk pemrograman total dan terintegrasi, yang memiliki instrumen untuk memutuskan informasi pengujian yang dapat menguji produk secara total dan juga dapat menemukan kesalahan. Biasanya, proses pengujian di dalam sebuah sistem dapat dilakukan dengan menerapkan 2 metode pengujian, yaitu *white box testing* dan *black box testing*.

1. *White Box Testing*

Proses pengujian sistem dengan cara mengamati modul untuk diperiksa dan dianalisis kode programnya. Jika *output* suatu sistem tidak cocok dengan keinginan, maka kode program akan di kompilasi lagi dan diperiksa lagi sampai sesuai dengan yang diinginkan. Dalam arti lain, *white box testing* menguji dengan melihat kode asli program di dalam sistem tanpa melihat tampilan *user interface* (UI). Pengujian dengan cara *white box testing* dilakukan dengan mengecek kode program asli setiap barisnya agar mengetahui penyebab *error* sehingga dapat segera diperbaiki.

2. *Black Box Testing*

Pengujian dengan cara *black box testing* dilakukan dengan cara melihat detail sistem, seperti tampilan UI nya, *tools* yang ada pada sistem, serta alur kerja sistem yang sesuai dengan keinginan pengguna. *Black box testing* lebih menguji hasil eksekusi tampilan UI agar nantinya mudah dipakai oleh *user*. Adapun beberapa pengujian utama pada *black box testing* dalam sistem ini, yaitu:

- a. Pengujian dalam menerapkan proses *resize* citra, konversi citra ke *grayscale* sampai penerapan *histogram equalization*.
- b. Pengujian dalam penerapan metode *High-Pass Filter* (HPF)
- c. Pengujian dalam penerapan metode *Fast Fourier Transform* (FFT) dalam proses *image enhancement*.
- d. Pengujian terhadap citra telapak tangan, dari proses *input*, *save* hingga sampai proses menghitung nilai *error*.

3.3.6. Implementasi/Penerapan

Tujuan di tahap implementasi dalam penelitian ini adalah untuk menghasilkan citra telapak tangan yang lebih baik, lebih jelas detail dan cirinya sehingga mudah diinterpretasikan oleh manusia serta citra telapak tangan yang telah di perbaiki kualitasnya yang nantinya dapat digunakan ketahap selanjutnya seperti di bidang pengenalan (*recognition*) objek. Serta hasil dari nilai *error* yang didapatkan, dapat menunjukkan diantara penerapan metode *high-pass filter* (HPF) dan *fast fourier transform* (FFT) manakah yang akan menghasilkan citra yang lebih baik (*better image*).

BAB IV

PEMBAHASAN DAN HASIL







4.1. Pembahasan

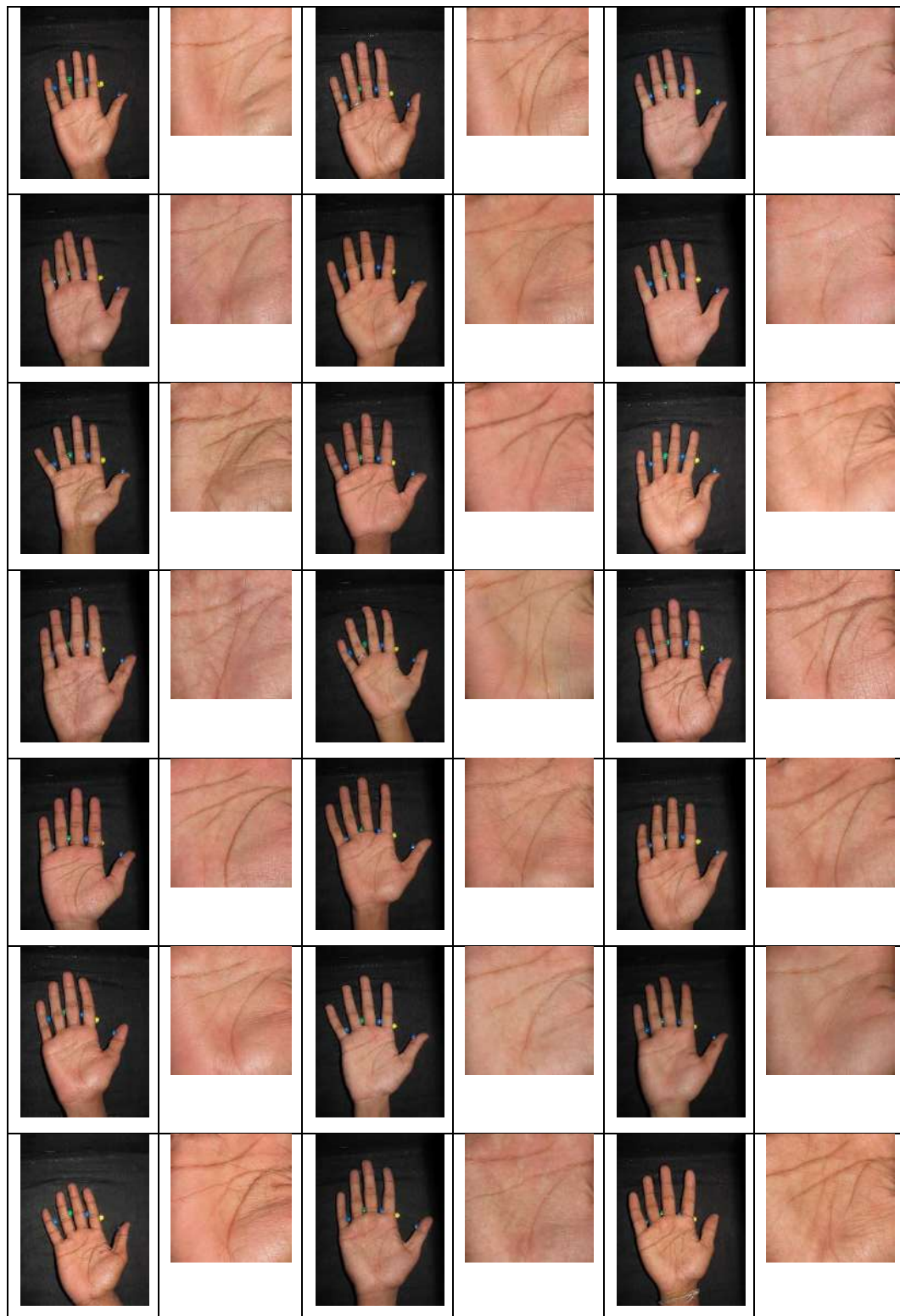
Adapun beberapa tahapan yang akan dijelaskan dalam penelitian ini yaitu analisis data, representasi data, hasil analisis data, dan perancangan sebagai berikut.

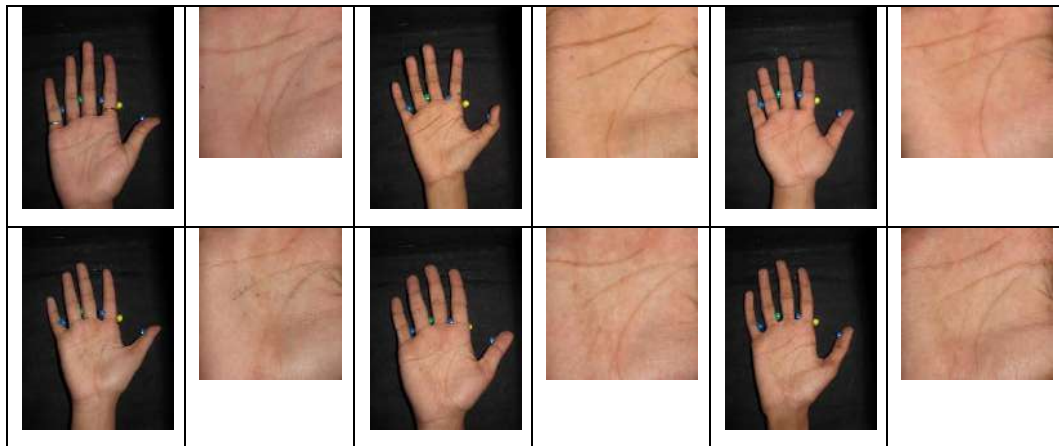
4.1.1. Analisis Data

Citra yang akan digunakan sebagai data dalam penelitian ini adalah berupa citra telapak tangan (*palmprint image*) normal (tidak kotor, berminyak ataupun terdapat bekas luka). Yang nantinya citra ini akan diolah kedalam dua tahap utama, yakni tahap *pre-processing* dan tahap *process*. Dimana pada tahap *pre-processing* akan dilakukan *resize* citra, konversi citra RGB ke *grayscale*, dan meningkatkan kontras dengan *histogram equalization*. Kemudian pada tahap *process*, akan menggunakan metode *high-pass filter* dan *fast fourier transform* dalam memperbaiki kualitas citra telapak tangan dengan menggunakan Matlab 2015a. Berikut ini merupakan tampilan dari citra telapak tangan yang akan digunakan:

Tabel 4.1. Database Citra Telapak Tangan

Citra Asli	Citra <i>Cropping</i>	Citra Asli	Citra <i>Cropping</i>	Citra Asli	Citra <i>Cropping</i>
					





Adapun langkah-langkah dalam program penelitian ini adalah:

- Menginput citra telapak tangan yang sebelumnya sudah di *cropping* terlebih dahulu dengan format citra *.bmp.
- Melakukan tahap *pre-processing* terhadap citra telapak tangan.
- Melakukan tahap *process* terhadap citra telapak tangan.
- Menghitung nilai *error* menggunakan MSE dan PSNR.

4.1.2. Representasi Data

Dalam penelitian ini, dibagi menjadi dua tahap yakni *pre-processing* dan *process*. Adapun beberapa tahapan dalam tahap *pre-processing* adalah sebagai berikut:

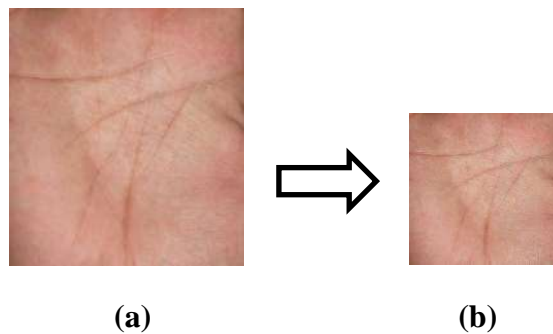
a. *Cropping*

Artinya memotong gambar yang akan digunakan dan menghilangkan gambar yang tidak terpakai. Untuk ukuran dan posisi gambar yang akan dipotong akan berubah tergantung dari gambar yang sebenarnya karena ukuran telapak tangan setiap orang sangatlah tak sama. Untuk citra hasil *cropping* dapat dilihat pada tabel 4.1, dimana citra telapak tangan asli dan lengkap beserta jari-jarinya di *crop* dan di ambil hanya bagian tengah telapak tangan saja.

b. *Resize*

Resize adalah suatu proses untuk mengubah ukuran piksel citra menjadi lebih besar ataupun lebih kecil dari citra sebelumnya. Setelah citra hasil *cropping* di *input* kesistem, selanjutnya citra akan di *resize* menjadi 256x256 piksel. Adapun sintaks *resize* pada Matlab adalah sebagai berikut:

```
>> B = imresize (A, [numrows numcols]);
```



Gambar: 4.1. (a) Citra Sebelum *Resize* dan (b) Citra *Resize* 256x256

c. *Grayscale*

Setelah citra di *resize*, selanjutnya citra akan di ubah dari citra RGB ke citra *grayscale*. Dengan menggunakan fungsi *rgb2gray* pada Matlab, citra RGB akan dikonversikan menjadi citra *grayscale*. Secara matematis untuk mencari nilai *grayscale* dengan fungsi *rgb2gray* pada Matlab adalah sebagai berikut:

$$\text{Grayscale} = 0.299 * R + 0.587 * G + 0.114 * B \dots\dots\dots (4.1)$$

Adapun sintaks untuk konversi ke citra *grayscale* pada Matlab adalah sebagai berikut:

```
>> G = rgb2gray (RGB);
```

Berikut ini adalah langkah-langkah untuk mendapatkan citra *grayscale* dengan sebagai berikut:

- Mendapatkan matriks RGB dari citra awal (sebelum di *grayscale*). Adapun matriksnya adalah sebagai berikut:

$$R = \begin{bmatrix} 178 & 195 & 205 & 208 & 190 \\ 193 & 208 & 214 & 213 & 190 \\ 199 & 207 & 214 & 211 & 197 \\ 205 & 209 & 207 & 207 & 200 \\ 191 & 215 & 207 & 207 & 197 \end{bmatrix} \quad G = \begin{bmatrix} 122 & 136 & 148 & 145 & 130 \\ 138 & 157 & 165 & 165 & 138 \\ 148 & 156 & 165 & 166 & 146 \\ 152 & 153 & 151 & 158 & 142 \\ 136 & 157 & 148 & 158 & 146 \end{bmatrix}$$

$$B = \begin{bmatrix} 103 & 116 & 130 & 129 & 114 \\ 118 & 135 & 146 & 147 & 118 \\ 129 & 134 & 145 & 145 & 126 \\ 135 & 133 & 130 & 140 & 126 \\ 121 & 141 & 128 & 141 & 130 \end{bmatrix}$$

- Mengubah matriks RGB menjadi matriks *grayscale* dengan menggunakan rumus sebelumnya. Dan berikut ini adalah perhitungan untuk mendapatkan matriks *grayscale*.

$$P1 = (0.299 * 178) + (0.587 * 122) + (0.114 * 103) = 53.222 + 71.614 + 11.742 = 136.578 = \mathbf{137}$$

$$P2 = (0.299 * 195) + (0.587 * 136) + (0.114 * 116) = 55.305 + 80.032 + 13.188 = 148.525 = \mathbf{151}$$

$$P3 = (0.299 * 205) + (0.587 * 148) + (0.114 * 130) = 61.295 + 86.876 + 14.82 = 163.0 = \mathbf{163}$$

$$P4 = (0.299 * 208) + (0.587 * 145) + (0.114 * 129) = 62.192 + 85.115 + 14.802 = 162.109 = \mathbf{162}$$

$$P5 = (0.299 * 190) + (0.587 * 130) + (0.114 * 114) = 56.81 + 76.31 + 12.996 = 146.116 = \mathbf{146}$$

$$P6 = (0.299 * 193) + (0.587 * 138) + (0.114 * 118) = 57.707 + 81.006 + 13.452 = 152.165 = \mathbf{152}$$

$$P7 = (0.299 * 208) + (0.587 * 157) + (0.114 * 135) = 62.192 + 92.259 + 15.39 = 169.841 = \mathbf{170}$$

$$P8 = (0.299 * 214) + (0.587 * 165) + (0.114 * 146) = 64.086 + 96.855 + 16.644 = 177.585 = \mathbf{177}$$

$$P9 = (0.299 * 213) + (0.587 * 165) + (0.114 * 147) = 63.687 + 96.855 + 16.758 = 177.3 = \mathbf{177}$$

$$P10 = (0.299 * 190) + (0.587 * 138) + (0.114 * 118) = 56.81 + 81.006 + 13.452 = 151.268 = \mathbf{151}$$

$$P11 = (0.299 * 199) + (0.587 * 148) + (0.114 * 129) = 59.501 + 86.876 + 14.802 = 161.179 = \mathbf{161}$$

$$P12 = (0.299 * 207) + (0.587 * 156) + (0.114 * 134) = 61.793 + 91.572 + 15.276 = 168.641 = \mathbf{169}$$

$$P13 = (0.299 * 214) + (0.587 * 165) + (0.114 * 145) = 64.086 + 96.855 + 16.53 = 177.471 = \mathbf{177}$$

$$P14 = (0.299 * 211) + (0.587 * 166) + (0.114 * 145) = 63.089 + 97.642 + 16.53 = 177.261 = \mathbf{177}$$

$$P15 = (0.299 * 197) + (0.587 * 146) + (0.114 * 126) = 58.903 + 85.702 + 14.424 = 159.029 = \mathbf{159}$$

$$P16 = (0.299 * 205) + (0.587 * 152) + (0.114 * 135) = 61.295 + 89.224 + 15.39 = 165.909 = \mathbf{166}$$

$$P17 = (0.299 * 209) + (0.587 * 153) + (0.114 * 133) = 61.581 + 89.811 + 15.172 = 166.564 = \mathbf{167}$$

$$P18 = (0.299 * 207) + (0.587 * 151) + (0.114 * 130) = 61.793 + 88.637 + 14.82 = 165.25 = \mathbf{165}$$

$$P19 = (0.299 * 207) + (0.587 * 158) + (0.114 * 140) = 61.793 + 92.746 + 15.96 = 170.599 = \mathbf{171}$$

$$P20 = (0.299 * 200) + (0.587 * 142) + (0.114 * 126) = 157.518 = \mathbf{158}$$

$$P21 = (0.299 * 191) + (0.587 * 136) + (0.114 * 121) = 150.735 = \mathbf{151}$$

$$P22 = (0.299 * 215) + (0.587 * 157) + (0.114 * 141) = 172.518 = \mathbf{173}$$

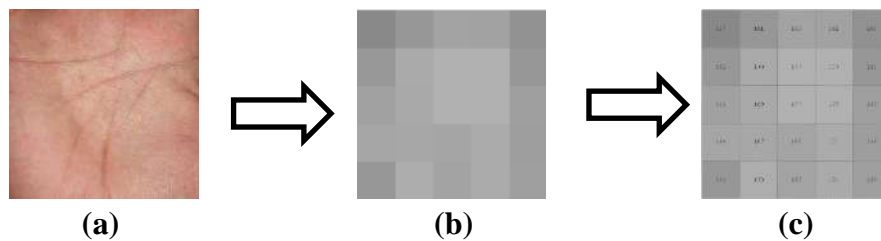
$$P23 = (0.299 * 207) + (0.587 * 148) + (0.114 * 128) = 163.361 = \mathbf{163}$$

$$P24 = (0.299 * 207) + (0.587 * 158) + (0.114 * 141) = 170.713 = \mathbf{171}$$

$$P25 = (0.299 * 197) + (0.587 * 146) + (0.114 * 130) = 159.425 = \mathbf{159}$$

- Maka, di dapatlah matriks *grayscale* seperti di bawah ini:

$$\begin{bmatrix} 137 & 151 & 163 & 162 & 146 \\ 152 & 170 & 177 & 177 & 151 \\ 161 & 169 & 177 & 177 & 159 \\ 166 & 167 & 165 & 171 & 158 \\ 151 & 173 & 163 & 171 & 159 \end{bmatrix}$$



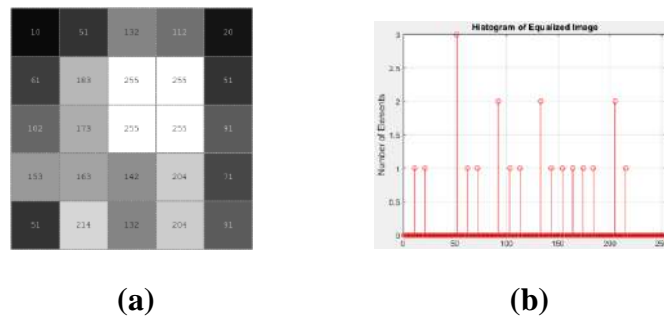
Gambar: 4.2. (a) Citra Asli 256x256, (b) Citra *Grayscale* 5x5 dan (c) *Region* piksel Citra 5x5

d. *Histogram Equalization*

Setelah citra *grayscale* didapat, tahap *pre-processing* selanjutnya yaitu dengan mengoptimalkan kontras citra memakai perataan histogram (*histogram equalization*). Adapun sintaks yang digunakan sebagai berikut:

```
>> J = histeq (I);
>> Imhist (J);
```

Dimana *histeq* digunakan untuk menampilkan citra histogram ekualisasi dan *imhist* untuk menampilkan histogram dari citra tersebut.



Gambar: 4.3. (a) *Region* Piksel Citra Histogram Ekualisasi, dan (b) Histogram Citra.

Adapun tahapan-tahapan untuk mendapatkan nilai histogram ekualisasi adalah sebagai berikut:

- Misalkan, dari matriks *grayscale* sebelumnya kita lakukan histogram ekualisasi.

$$\begin{bmatrix} 137 & 151 & 163 & 162 & 146 \\ 152 & 170 & 177 & 177 & 151 \\ 161 & 169 & 177 & 177 & 159 \\ 166 & 167 & 165 & 171 & 158 \\ 151 & 173 & 163 & 171 & 159 \end{bmatrix}$$

- Dari nilai piksel pada matriks *grayscale* tersebut, kita tentukan piksel *gray level* (j), frekuensinya (f), dan distribusi kumulatifnya (cd) kedalam bentuk tabel berikut.

Tabel 4.2. Nilai Frekuensi dan Distribusi Kumulatif Histogram Ekualisasi

<i>Gray Level</i>	F	$CD = \sum_{j=0}^k n_j$	<i>Gray Level</i>	F	$CD = \sum_{j=0}^k n_j$	<i>Gray Level</i>	F	$CD = \sum_{j=0}^k n_j$
137	1	1	161	1	10	169	1	17
146	1	2	162	1	11	170	1	18
151	3	5	163	2	13	171	2	20
152	1	6	165	1	14	173	1	21
158	1	7	166	1	15	177	4	25
159	2	9	167	1	16			

- Selanjutnya, menghitung nilai keabuan hasil histogram ekualisasi dari perhitungan distribusi kumulatif, dengan menggunakan rumus berikut:

$$S = CD(i) * \frac{256-1}{N} \dots\dots\dots (4.2)$$

$$j1 = CD(1) * \frac{256-1}{N} = 1 * \frac{255}{25} = 10.2 = \mathbf{10}$$

$$j2 = CD(2) * \frac{256-1}{N} = 2 * \frac{255}{25} = 20.4 = \mathbf{20}$$

$$j3 = CD(3) * \frac{256-1}{N} = 5 * \frac{255}{25} = \mathbf{51}$$

$$j4 = CD(4) * \frac{256-1}{N} = 6 * \frac{255}{25} = 61.2 = \mathbf{61}$$

$$j5 = CD(5) * \frac{256-1}{N} = 7 * \frac{255}{25} = 71.4 = \mathbf{71}$$

$$j6 = CD(6) * \frac{256-1}{N} = 9 * \frac{255}{25} = 91.2 = \mathbf{91}$$

$$j7 = CD(7) * \frac{256-1}{N} = 10 * \frac{255}{25} = \mathbf{102}$$

$$j8 = CD(8) * \frac{256-1}{N} = 11 * \frac{255}{25} = 112.2 = \mathbf{112}$$

$$j9 = CD(9) * \frac{256-1}{N} = 13 * \frac{255}{25} = 132.6 = \mathbf{133}$$

$$j10 = CD(10) * \frac{256-1}{N} = 14 * \frac{255}{25} = 142.8 = \mathbf{143}$$

$$j11 = CD(11) * \frac{256-1}{N} = 15 * \frac{255}{25} = \mathbf{153}$$

$$j12 = CD(12) * \frac{256-1}{N} = 16 * \frac{255}{25} = 163.2 = \mathbf{163}$$

$$j13 = CD(13) * \frac{256-1}{N} = 17 * \frac{255}{25} = 173.4 = \mathbf{173}$$

$$j14 = CD(14) * \frac{256-1}{N} = 18 * \frac{255}{25} = 183.6 = \mathbf{184}$$

$$j15 = CD(15) * \frac{256-1}{N} = 20 * \frac{255}{25} = \mathbf{204}$$

$$j16 = CD(21) * \frac{256-1}{N} = 21 * \frac{255}{25} = 214.2 = \mathbf{214}$$

$$j17 = CD(17) * \frac{256 - 1}{N} = 25 * \frac{255}{25} = 255$$

- Selanjutnya, di dapatlah hasil perhitungan skala keabuan yang baru yang dapat ditinjau pada tabel berikut:

Tabel 4.3. Hasil *Gray Level* Histogram Ekualisasi

<i>Gray Level</i> (Awal)	<i>f</i>	<i>CD</i>	<i>Gray Level</i> (Hasil)	<i>Gray Level</i> (awal)	<i>f</i>	<i>CD</i>	<i>Gray Level</i> (Hasil)
137	1	1	10	165	1	14	143
146	1	2	20	166	1	15	153
151	3	5	51	167	1	16	163
152	1	6	61	169	1	17	173
158	1	7	71	170	1	18	184
159	2	9	91	171	2	20	204
161	1	10	102	173	1	21	214
162	1	11	112	177	4	25	255
163	2	13	133				

- Maka didapatkan matriks histogram ekualisasi, dengan mengganti nilai piksel awal menjadi nilai piksel akhir. Dan dibawah ini adalah matriks dari hasil histogram ekualisasi pada citra *grayscale*.

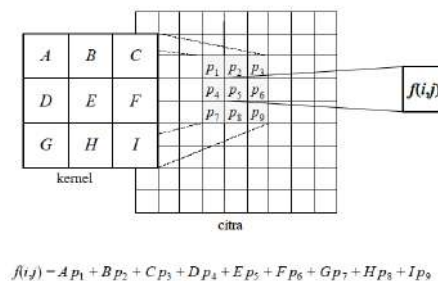
$$\begin{bmatrix} 10 & 51 & 132 & 112 & 20 \\ 61 & 183 & 255 & 255 & 51 \\ 102 & 173 & 255 & 255 & 91 \\ 153 & 163 & 142 & 204 & 71 \\ 51 & 214 & 132 & 204 & 91 \end{bmatrix}$$

Setelah citra dilakukan proses *pre-processing*, selanjutnya citra telapak tangan akan di *process* menggunakan metode HPF dan FFT. Dan berikut akan dijelaskan penggunaan kedua metode tersebut:

e. *High-Pass Filter* (HPF)

Tapis lolos tinggi (*high-pass filter*) merupakan salah satu metode berdomain spasial. Teknik pemrosesan sebuah citra yang berdasarkan metode spasial beroperasi langsung pada pikselnya dengan operasi konvolusi. Konvolusi dapat diterapkan pada proses penajaman citra, dengan mengalikan antara matriks citra dan kernel *filter*. Konvolusi pada 2 buah fungsi $f(x)$ dan $g(x)$ dapat didefinisikan kedalam persamaan berikut:

$$h(x) = f(x) * g(x) = \int_{-\infty}^{\infty} f(a) g(x - a) da \dots \dots \dots (4.3)$$

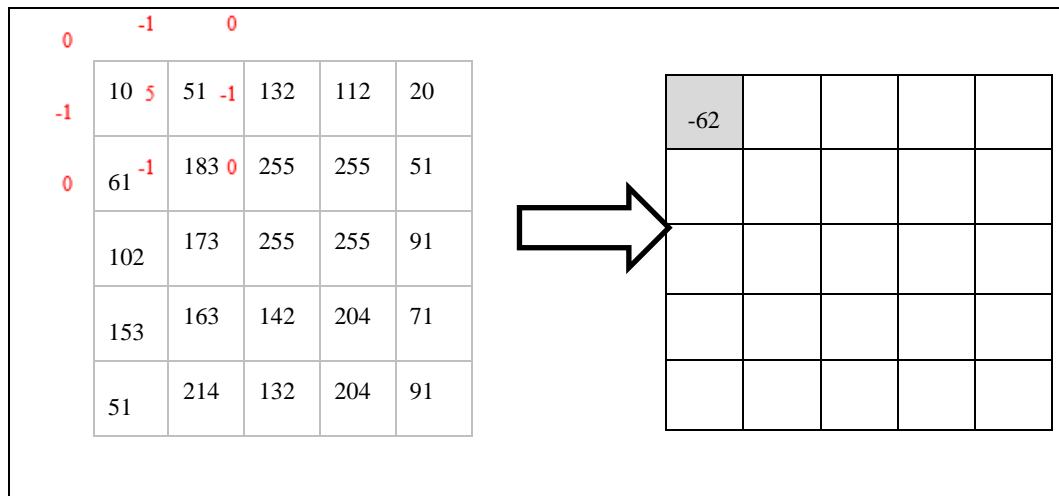


Gambar: 4.4. Proses Konvolusi Terhadap Citra (Munir, 2004)

$$f(x, y) = \begin{bmatrix} 10 & 51 & 132 & 112 & 20 \\ 61 & 183 & 255 & 255 & 51 \\ 102 & 173 & 255 & 255 & 91 \\ 153 & 163 & 142 & 204 & 71 \\ 51 & 214 & 132 & 204 & 91 \end{bmatrix} \quad g(x, y) = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

Misal, $f(x, y)$ didefinisikan sebagai matriks citra input dari hasil histogram ekualisasi dengan ukuran 5x5 dan $g(x, y)$ adalah sebagai kernel *filter* HPF dengan $K= 1$. Maka, operasi konvolusi *neighbor* dari citra $f(x, y)$ dan filter $g(x, y)$ adalah sebagai berikut:

- Tempatkan posisi nilai tengah kernel filter pada sudut kiri atas, lalu kalikan antara matriks $f(x, y) * g(x, y)$:



Maka, nilai intensitas baru dari piksel hasil konvolusi pada posisi (0,0) dari kernel adalah:

$$P1 = (10 * 5) + (51 * -1) + (61 * -1) + (183 * 0) = 50 - 51 - 61 + 0 \\ = -62$$

- Selanjutnya, geser kernel satu piksel kekanan dan hitung nilai piksel pada posisi (0,0) seperti berikut:

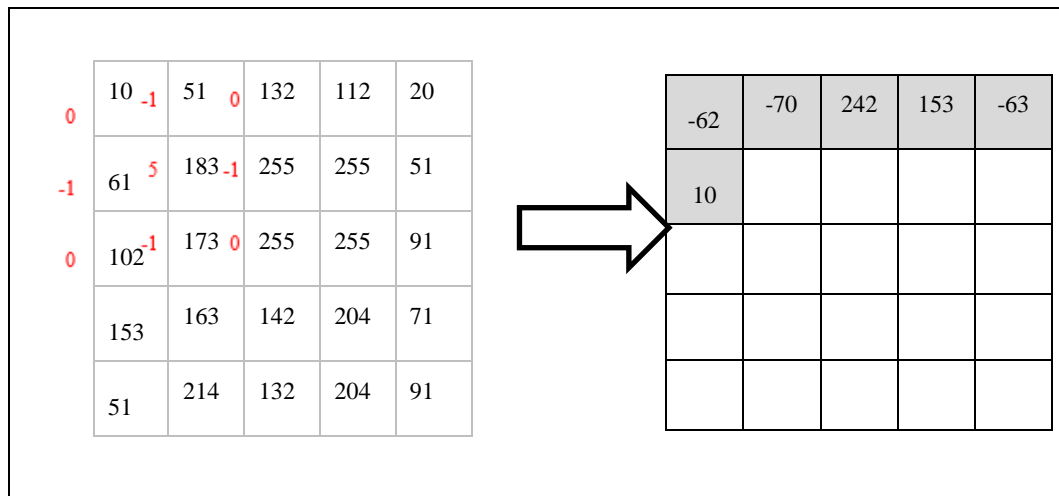
$$P2 = (10 * -1) + (51 * 5) + (132 * -1) + (61 * 0) + (183 * -1) + (255 * 0) \\ = -10 + 255 - 132 + 0 - 183 + 0 = -70$$

$$P3 = (51 * -1) + (132 * 5) + (112 * -1) + (183 * 0) + (255 * -1) \\ + (255 * 0) = 242$$

$$P4 = (132 * -1) + (112 * 5) + (20 * -1) + (255 * 0) + (255 * -1) \\ + (51 * 0) = 153$$

$$P5 = (112 * -1) + (20 * 5) + (255 * 0) + (51 * -1) = -63$$

- Selanjutnya, geser kernel filter satu piksel ke bawah dan hitung nilai piksel dari posisi kiri pada posisi (0,0) seperti berikut:



$$P6 = (10 * -1) + (51 * 0) + (61 * 5) + (183 * -1) + (102 * -1) + (173 * 0) \\ = \mathbf{10}$$

- Geser kernel satu piksel ke kanan lagi, lalu hitung nilai piksel pada posisi (0,0) seperti berikut:

$$P7 = (10 * 0) + (51 * -1) + (132 * 0) + (61 * -1) + (183 * 5) + (255 * -1) \\ + (102 * 0) + (173 * -1) + (255 * 0) = \mathbf{385}$$

$$P8 = (51 * 0) + (132 * -1) + (112 * 0) + (183 * -1) + (255 * 5) \\ + (255 * -1) + (173 * 0) + (255 * -1) + (255 * 0) = \mathbf{450}$$

$$P9 = (132 * 0) + (112 * -1) + (20 * 0) + (255 * -1) + (255 * 5) \\ + (51 * -1) + (255 * 0) + (255 * -1) + (91 * 0) = \mathbf{602}$$

$$P10 = (112 * 0) + (20 * -1) + (255 * -1) + (51 * 5) + (255 * 0) \\ + (91 * -1) = \mathbf{-111}$$

$$P11 = (61 * -1) + (183 * 0) + (102 * 5) + (173 * -1) + (153 * -1) \\ + (163 * 0) = \mathbf{123}$$

$$P12 = (61 * 0) + (183 * -1) + (255 * 0) + (102 * -1) + (173 * 5) \\ + (255 * -1) + (153 * 0) + (163 * -1) + (142 * 0) = \mathbf{162}$$

$$P13 = (183 * 0) + (255 * -1) + (255 * 0) + (173 * -1) + (255 * 5) \\ + (255 * -1) + (163 * 0) + (142 * -1) + (204 * 0) = \mathbf{450}$$

$$P14 = (255 * 0) + (255 * -1) + (51 * 0) + (255 * -1) + (255 * 5) \\ + (91 * -1) + (142 * 0) + (204 * -1) + (71 * 0) = \mathbf{470}$$

$$P15 = (255 * 0) + (51 * -1) + (255 * -1) + (91 * 5) + (204 * 0) \\ + (71 * -1) = \mathbf{78}$$

$$P16 = (102 * -1) + (173 * 0) + (153 * 5) + (163 * -1) + (51 * -1) \\ + (214 * 0) = \mathbf{449}$$

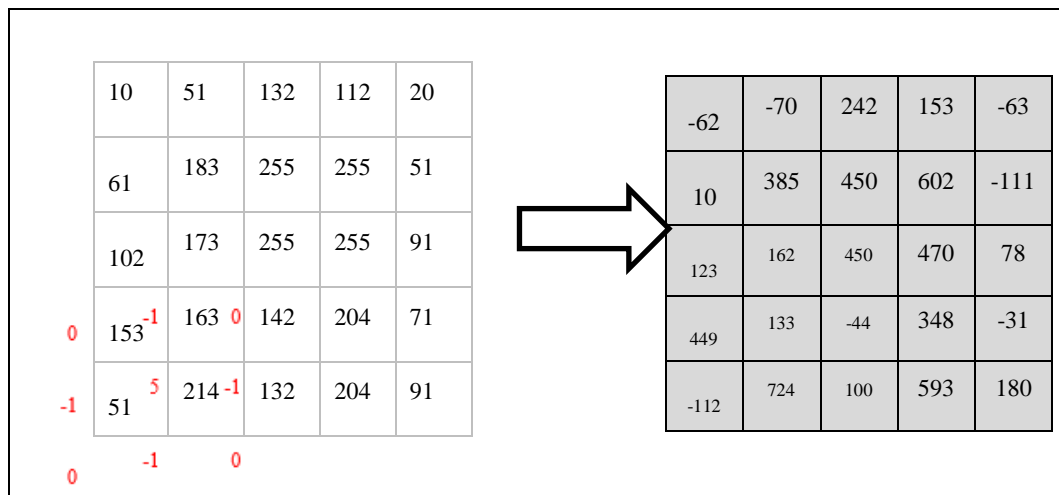
$$P17 = (102 * 0) + (173 * -1) + (255 * 0) + (153 * -1) + (163 * 5) \\ + (142 * -1) + (51 * 0) + (214 * -1) + (132 * 0) = \mathbf{133}$$

$$P18 = (173 * 0) + (255 * -1) + (255 * 0) + (163 * -1) + (142 * 5) \\ + (204 * -1) + (214 * 0) + (132 * -1) + (204 * 0) = \mathbf{-44}$$

$$P19 = (255 * 0) + (255 * -1) + (91 * 0) + (142 * -1) + (204 * 5) \\ + (71 * -1) + (132 * 0) + (204 * -1) + (91 * 0) = \mathbf{348}$$

$$P20 = (255 * 0) + (91 * -1) + (204 * -1) + (71 * 5) + (204 * 0) \\ + (91 * -1) = \mathbf{-31}$$

- Selanjutnya, geser kernel filter satu piksel ke bawah dan hitung nilai piksel dari posisi kiri dan dilanjutkan sampai piksel terakhir, pada posisi (0,0) seperti berikut:



$$P21 = (153 * -1) + (163 * 0) + (51 * 5) + (214 * -1) = \mathbf{-112}$$

$$P22 = (153 * 0) + (163 * -1) + (142 * 0) + (51 * -1) + (214 * 5) + (132 \\ * -1) = \mathbf{724}$$

$$P23 = (163 * 0) + (142 * -1) + (204 * 0) + (214 * -1) + (132 * 5) + (204 \\ * -1) = \mathbf{100}$$

$$P24 = (142 * 0) + (204 * -1) + (71 * 0) + (132 * -1) + (204 * 5) + (91 \\ * -1) = \mathbf{593}$$

$$P_{25} = (204 * 0) + (71 * -1) + (204 * -1) + (91 * 5) = \mathbf{180}$$

Dikarenakan nilai dari hasil konvolusi menghasilkan nilai negatif dan nilai piksel yang paling tinggi dari nilai keabuan tertinggi, maka hasil konvolusi yang bernilai negatif diubah menjadi 0 dan hasil konvolusi yang melebihi nilai keabuan maksimum diubah menjadi 255 (Munir, 2004).

-62	-70	242	153	-63		0	0	242	153	0
10	385	450	602	-111		10	255	255	255	0
123	162	450	470	78		123	162	255	255	78
449	133	-44	348	-31		255	133	0	255	0
-112	724	100	593	180		0	255	100	255	180

0	0	242	153	0
10	255	255	255	0
123	162	255	255	78
255	133	0	255	0
0	255	100	255	180

Gambar: 4.5. Representasi Piksel Hasil HPFdengan K=1

f. *Fast Fourier Transform* (FFT)

Ada beberapa langkah dalam memproses citra dengan menggunakan FFT, diantaranya adalah sebagai berikut:

1. *Transformasi Fourier*

Pada tahapan ini, yang akan dilakukan adalah dengan mencari nilai matriks transformasi fourier dari citra *input*. Dimana citra yang digunakan adalah citra hasil *pre-processing* yaitu citra histogram ekualisasi.

$$\begin{bmatrix} 10 & 51 & 132 & 112 & 20 \\ 61 & 183 & 255 & 255 & 51 \\ 102 & 173 & 255 & 255 & 91 \\ 153 & 163 & 142 & 204 & 71 \\ 51 & 214 & 132 & 204 & 91 \end{bmatrix}$$

$$F(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \left(\cos \left(2\pi \left(\frac{ux}{M} + \frac{vy}{N} \right) \right) - j \sin \left(2\pi \left(\frac{ux}{M} + \frac{vy}{N} \right) \right) \right) \dots (4.4)$$

Dimana:

M = baris

N = kolom

u = frekuensi bernilai dari 0 sampai dengan M-1

v = frekuensi bernilai dari 0 sampai dengan N-1

x = koordinat baris piksel

y = koordinat kolom piksel

π = 180

$F(u, v)$ = koefisien fourier

Rumus diatas akan digunakan dalam mencari nilai transformasi fourier dari citra, adapun langkah-langkah perhitungan manual nya adalah sebagai berikut:

$$\begin{aligned} F(0, 0) &= 10 \begin{bmatrix} \cos \left(2 \cdot 180 \left(\frac{0.0}{5} + \frac{0.0}{5} \right) \right) \\ -j \sin \left(2 \cdot 180 \left(\frac{0.0}{5} + \frac{0.0}{5} \right) \right) \end{bmatrix} + 51 \begin{bmatrix} \cos \left(2 \cdot 180 \left(\frac{0.0}{5} + \frac{0.1}{5} \right) \right) \\ -j \sin \left(2 \cdot 180 \left(\frac{0.0}{5} + \frac{0.1}{5} \right) \right) \end{bmatrix} \\ &+ 132 \begin{bmatrix} \cos \left(2 \cdot 180 \left(\frac{0.0}{5} + \frac{0.2}{5} \right) \right) \\ -j \sin \left(2 \cdot 180 \left(\frac{0.0}{5} + \frac{0.2}{5} \right) \right) \end{bmatrix} + 112 \begin{bmatrix} \cos \left(2 \cdot 180 \left(\frac{0.0}{5} + \frac{0.3}{5} \right) \right) \\ -j \sin \left(2 \cdot 180 \left(\frac{0.0}{5} + \frac{0.3}{5} \right) \right) \end{bmatrix} \\ &+ 20 \begin{bmatrix} \cos \left(2 \cdot 180 \left(\frac{0.0}{5} + \frac{0.4}{5} \right) \right) \\ -j \sin \left(2 \cdot 180 \left(\frac{0.0}{5} + \frac{0.4}{5} \right) \right) \end{bmatrix} + \dots \\ &+ 51 \begin{bmatrix} \cos \left(2 \cdot 180 \left(\frac{0.4}{5} + \frac{0.0}{5} \right) \right) \\ -j \sin \left(2 \cdot 180 \left(\frac{0.4}{5} + \frac{0.0}{5} \right) \right) \end{bmatrix} + 214 \begin{bmatrix} \cos \left(2 \cdot 180 \left(\frac{0.4}{5} + \frac{0.1}{5} \right) \right) \\ -j \sin \left(2 \cdot 180 \left(\frac{0.4}{5} + \frac{0.1}{5} \right) \right) \end{bmatrix} \end{aligned}$$

$$\begin{aligned}
& +132 \begin{bmatrix} \cos \left(2*180 \left(\frac{0.4}{5} + \frac{0.2}{5} \right) \right) \\ -j \sin \left(2*180 \left(\frac{0.4}{5} + \frac{0.2}{5} \right) \right) \end{bmatrix} + 204 \begin{bmatrix} \cos \left(2*180 \left(\frac{0.4}{5} + \frac{0.3}{5} \right) \right) \\ -j \sin \left(2*180 \left(\frac{0.4}{5} + \frac{0.3}{5} \right) \right) \end{bmatrix} \\
& + 91 \begin{bmatrix} \cos \left(2*180 \left(\frac{0.4}{5} + \frac{0.4}{5} \right) \right) \\ -j \sin \left(2*180 \left(\frac{0.4}{5} + \frac{0.4}{5} \right) \right) \end{bmatrix}
\end{aligned}$$

$$F(\mathbf{0}, \mathbf{0}) = 10[\cos(0) - j \sin(0)] + 51[\cos(0) - j \sin(0)] + 132[\cos(0) - j \sin(0)]$$

$$+ 112[\cos(0) - j \sin(0)] + 20[\cos(0) - j \sin(0)] + \dots$$

$$+ 51[\cos(0) - j \sin(0)] + 214[\cos(0) - j \sin(0)] + 132[\cos(0) - j \sin(0)]$$

$$+ 204[\cos(0) - j \sin(0)] + 91[\cos(0) - j \sin(0)]$$

$$F(\mathbf{0}, \mathbf{0}) = 10[1 - 0] + 51[1 - 0] + 132[1 - 0] + 112[1 - 0]$$

$$+ 20[1 - 0] + \dots$$

$$+ 51[1 - 0] + 214[1 - 0] + 132[1 - 0] + 204[1 - 0] + 91[1 - 0]$$

$$F(\mathbf{0}, \mathbf{0}) = 10 + 51 + 132 + 112 + 20 + \dots + 51 + 214 + 132 + 204$$

$$+ 91$$

$$F(\mathbf{0}, \mathbf{0}) = 3341$$

$$\begin{aligned}
F(\mathbf{0}, \mathbf{1}) = & 10 \begin{bmatrix} \cos \left(2*180 \left(\frac{0.0}{5} + \frac{1.0}{5} \right) \right) \\ -j \sin \left(2*180 \left(\frac{0.0}{5} + \frac{1.0}{5} \right) \right) \end{bmatrix} + 51 \begin{bmatrix} \cos \left(2*180 \left(\frac{0.0}{5} + \frac{1.1}{5} \right) \right) \\ -j \sin \left(2*180 \left(\frac{0.0}{5} + \frac{1.1}{5} \right) \right) \end{bmatrix}
\end{aligned}$$

$$\begin{aligned}
& + 132 \begin{bmatrix} \cos \left(2*180 \left(\frac{0.0}{5} + \frac{1.2}{5} \right) \right) \\ -j \sin \left(2*180 \left(\frac{0.0}{5} + \frac{1.2}{5} \right) \right) \end{bmatrix} + 112 \begin{bmatrix} \cos \left(2*180 \left(\frac{0.0}{5} + \frac{1.3}{5} \right) \right) \\ -j \sin \left(2*180 \left(\frac{0.0}{5} + \frac{1.3}{5} \right) \right) \end{bmatrix}
\end{aligned}$$

$$\begin{aligned}
& + 20 \begin{bmatrix} \cos \left(2*180 \left(\frac{0.0}{5} + \frac{1.4}{5} \right) \right) \\ -j \sin \left(2*180 \left(\frac{0.0}{5} + \frac{1.4}{5} \right) \right) \end{bmatrix} + \dots
\end{aligned}$$

$$\begin{aligned}
& + 51 \begin{bmatrix} \cos \left(2*180 \left(\frac{0.4}{5} + \frac{1.0}{5} \right) \right) \\ -j \sin \left(2*180 \left(\frac{0.4}{5} + \frac{1.0}{5} \right) \right) \end{bmatrix} + 214 \begin{bmatrix} \cos \left(2*180 \left(\frac{0.4}{5} + \frac{1.1}{5} \right) \right) \\ -j \sin \left(2*180 \left(\frac{0.4}{5} + \frac{1.1}{5} \right) \right) \end{bmatrix}
\end{aligned}$$

$$\begin{aligned}
& +132 \left[\cos \left(2 \cdot 180 \left(\frac{0.4}{5} + \frac{1.2}{5} \right) \right) \right] + 204 \left[\cos \left(2 \cdot 180 \left(\frac{0.4}{5} + \frac{1.3}{5} \right) \right) \right] \\
& -j \sin \left(2 \cdot 180 \left(\frac{0.4}{5} + \frac{1.2}{5} \right) \right) -j \sin \left(2 \cdot 180 \left(\frac{0.4}{5} + \frac{1.3}{5} \right) \right) \\
& +91 \left[\cos \left(2 \cdot 180 \left(\frac{0.4}{5} + \frac{1.4}{5} \right) \right) \right] \\
& -j \sin \left(2 \cdot 180 \left(\frac{0.4}{5} + \frac{1.4}{5} \right) \right)
\end{aligned}$$

$$\begin{aligned}
F(\mathbf{0}, \mathbf{1}) &= 10[\cos(0) - j \sin(0)] + 51 \left[\cos\left(\frac{2\pi}{5}\right) - j \sin\left(\frac{2\pi}{5}\right) \right] \\
&+ 132 \left[\cos\left(\frac{4\pi}{5}\right) - j \sin\left(\frac{4\pi}{5}\right) \right] \\
&+ 112 \left[\cos\left(\frac{6\pi}{5}\right) - j \sin\left(\frac{6\pi}{5}\right) \right] + 20 \left[\cos\left(\frac{8\pi}{5}\right) - j \sin\left(\frac{8\pi}{5}\right) \right] + \dots \\
&+ 51[\cos(0) - j \sin(0)] + 214 \left[\cos\left(\frac{2\pi}{5}\right) - j \sin\left(\frac{2\pi}{5}\right) \right] + 132 \left[\cos\left(\frac{4\pi}{5}\right) - j \sin\left(\frac{4\pi}{5}\right) \right] \\
&+ 204 \left[\cos\left(\frac{6\pi}{5}\right) - j \sin\left(\frac{6\pi}{5}\right) \right] + 91 \left[\cos\left(\frac{8\pi}{5}\right) - j \sin\left(\frac{8\pi}{5}\right) \right]
\end{aligned}$$

$$\begin{aligned}
F(\mathbf{0}, \mathbf{1}) &= 10[1 - 0] + 51[\cos(72) - j \sin(72)] + 132[\cos(144) - j \sin(144)] \\
&+ 112[\cos(216) - j \sin(216)] + 20[\cos(288) - j \sin(288)] + \dots \\
&+ 51[1 - 0] + 214[\cos(72) - j \sin(72)] + 132[\cos(144) - j \sin(144)] \\
&+ 204[\cos(216) - j \sin(216)] + 91[\cos(288) - j \sin(288)]
\end{aligned}$$

$$\begin{aligned}
F(\mathbf{0}, \mathbf{1}) &= 10[1 - 0] + 51[0,309 - 0,951 i] + 132[-0,809 - 0,587 i] \\
&+ 112[-0,809 - (-0,587) i] + 20[0,309 - (-0,951) i] + \dots \\
&+ 51[1 - 0] + 214[0,309 - 0,951 i] + 132[-0,809 - 0,587 i] \\
&+ 204[-0,809 - (-0,587) i] + 91[0,309 - (-0,951) i]
\end{aligned}$$

$$\begin{aligned}
F(\mathbf{0}, \mathbf{1}) &= 10 + (15,759 - 48,501 i) + (-106,788 - 77,484 i) \\
&+ (-90,608 - (-65,744 i)) + (6,18 - (-19,02 i)) + \dots \\
&+ 51 + (66,126 - 203,514 i) + (-106,788 - 77,484 i) \\
&+ (-165,036 - (-119,748 i)) + (8,119 - (-86,541 i))
\end{aligned}$$

$$F(\mathbf{0}, \mathbf{1}) = (-854,958 - 370,477 i)$$

$$\begin{aligned}
F(\mathbf{0}, \mathbf{2}) &= 10 + (-41,259 - 29,977 i) + (40,790 - (-125,539 i)) \\
&+ (34,609 - 106,518 i) + (-16,180 - (-11,755 i)) + \dots \\
&+ 51 + (-173,129 - 125,786 i) + (40,790 - (-125,539 i)) \\
&+ (63,039 - 194,015 i) + (-73,620 - (-53,488 i))
\end{aligned}$$

$$F(\mathbf{0}, \mathbf{2}) = (81,958 - 378,801 i)$$

$$\begin{aligned}
F(0,3) &= 10 + (-41,259 - (-29,977 i)) + (40,790 - 125,539 i) \\
&+ (34,609 - (-106,518 i)) + (-16,180 - 11,755 i) + \dots \\
&+ 51 + (-173,129 - (-125,786 i)) + (40,790 - 125,539 i) + \\
&+ (63,039 - (-194,015 i)) + (-73,620 - 53,488 i)
\end{aligned}$$

$$F(0,3) = (81,958 + 378,801 i)$$

$$\begin{aligned}
F(0,4) &= 10 + (15,759 - (-48,503 i)) + (-106,790 - (-77,587 i)) \\
&+ (-90,609 - 65,831 i) + (6,180 - 19,021 i) + \dots \\
&+ 51 + (66,129 - (-203,526 i)) + (-106,790 - (-77,587 i)) \\
&+ (-165,039 - 119,908 i) + (28,120 - 86,546 i)
\end{aligned}$$

$$F(0,4) = (-854,958 + 370,477 i)$$

$$F(1,0) = -514,109 - 191,522 i$$

$$F(1,1) = -125,640 + 249,042 i$$

$$F(1,2) = 12,458 - 37,053 i$$

$$F(1,3) = -183,955 - 164,590 i$$

$$F(1,4) = 2,799 + 246,456 i$$

$$F(2,0) = -388,890 + 69,581 i$$

$$F(2,1) = 70,955 + 4,737 i$$

$$F(2,2) = 90,140 + 32,998 i$$

$$F(2,3) = 139,200 - 222,264 i$$

$$F(2,4) = 79,541 - 156,961 i$$

$$F(3,0) = -388,890 - 69,581 i$$

$$F(3,1) = 79,541 + 156,961 i$$

$$F(3,2) = 139,200 + 222,264 i$$

$$F(3,3) = 90,140 - 32,998 i$$

$$F(3,4) = 70,955 - 4,737 i$$

$$F(4,0) = -514,109 + 191,522 i$$

$$F(4,1) = 2,799 - 246,456 i$$

$$F(4,2) = -183,955 + 164,590 i$$

$$F(4,3) = 12,458 + 37,053 i$$

$$F(4,4) = -125,640 - 249,042 i$$

3341	-854,958 - 370,477 <i>i</i>	81,958 - 378,801 <i>i</i>	81,958 + 378,801 <i>i</i>	-854,958 + 370,477 <i>i</i>
-514,109 - 191,522 <i>i</i>	-125,640 + 249,042 <i>i</i>	12,458 - 37,053 <i>i</i>	-183,955 - 164,590 <i>i</i>	2,799 + 246,456 <i>i</i>
-388,890 + 69,581 <i>i</i>	70,955 + 4,737 <i>i</i>	90,140 + 32,998 <i>i</i>	139,200 - 222,264 <i>i</i>	79,541 - 156,961 <i>i</i>
-388,890 - 69,581 <i>i</i>	79,541 + 156,961 <i>i</i>	139,200 + 222,264 <i>i</i>	90,140 - 32,998 <i>i</i>	70,955 - 4,737 <i>i</i>
-514,109 + 191,522 <i>i</i>	2,799 - 246,456 <i>i</i>	-183,955 + 164,590 <i>i</i>	12,458 + 37,053 <i>i</i>	-125,640 - 249,042 <i>i</i>

2. Filter Domain Frekuensi (IHPF, BHPF, GHPF)

Berikut ini adalah perhitungan manual menggunakan *filter* domain frekuensi yakni *Ideal High-Pass Filter*, *Butterworth High-Pass Filter*, dan *Gaussian High-Pass Filter*:

- Sebelumnya, tentukan terlebih dahulu nilai D_0 (frekuensi *Cut-off*) yang akan digunakan, misalnya $D_0 = 20$
- Selanjutnya tentukan nilai jarak dari titik (u, v) ke pusat frekuensi dengan menggunakan persamaan berikut:

$$D(u, v) =$$

$$\sqrt{u^2 + v^2} \dots\dots\dots (4.5)$$

$$D(0, 0) = \sqrt{0^2 + 0^2} = 0$$

$$D(0, 1) = \sqrt{0^2 + 1^2} = 1$$

$$D(0, 2) = \sqrt{0^2 + 2^2} = 2$$

$$D(0, 3) = \sqrt{0^2 + 3^2} = 3$$

$$D(0, 4) = \sqrt{0^2 + 4^2} = 4$$

$$D(1, 0) = \sqrt{1^2 + 0^2} = 1$$

$$D(1, 1) = \sqrt{1^2 + 1^2} = 1,41$$

$$D(1, 2) = \sqrt{1^2 + 2^2} = 2,23$$

$$D(1, 3) = \sqrt{1^2 + 3^2} = 3,16$$

$$D(1,4) = \sqrt{1^2 + 4^2} = 4,12$$

$$D(2,0) = \sqrt{2^2 + 0^2} = 2$$

$$D(2,1) = \sqrt{2^2 + 1^2} = 2,23$$

$$D(2,2) = \sqrt{2^2 + 2^2} = 2,82$$

$$D(2,3) = \sqrt{2^2 + 3^2} = 3,6$$

$$D(2,4) = \sqrt{2^2 + 4^2} = 4,47$$

$$D(3,0) = \sqrt{3^2 + 0^2} = 3$$

$$D(3,1) = \sqrt{3^2 + 1^2} = 3,16$$

$$D(3,2) = \sqrt{3^2 + 2^2} = 3,6$$

$$D(3,3) = \sqrt{3^2 + 3^2} = 4,24$$

$$D(3,4) = \sqrt{3^2 + 4^2} = 5$$

$$D(4,0) = \sqrt{4^2 + 0^2} = 4$$

$$D(4,1) = \sqrt{4^2 + 1^2} = 4,12$$

$$D(4,2) = \sqrt{4^2 + 2^2} = 4,47$$

$$D(4,3) = \sqrt{4^2 + 3^2} = 5$$

$$D(4,4) = \sqrt{4^2 + 4^2} = 5,65$$

- Lakukan perhitungan manual untuk setiap filter HPF domain frekuensi untuk mendapatkan nilai matriks setiap filter. Pertama, *Ideal High-Pass Filter* (IHPF) dengan menggunakan persamaan 2.12, didapatlah nilai matriks sebagai berikut:

Ideal High-Pass Filter (IHPF)

$$H(0,0) \leq D_0 = 0 \leq 20 = 0$$

$$H(0,1) \leq D_0 = 1 \leq 20 = 0$$

$$H(0,2) \leq D_0 = 2 \leq 20 = 0$$

$$H(0,3) \leq D_0 = 3 \leq 20 = 0$$

$$H(0,4) \leq D_0 = 4 \leq 20 = 0$$

$$H(1,0) \leq D_0 = 1 \leq 20 = 0$$

$$H(1,1) \leq D_0 = 1,41 \leq 20 = 0$$

$$H(1,2) \leq D_0 = 2,23 \leq 20 = 0$$

$$H(1,3) \leq D_0 = 3,16 \leq 20 = 0$$

$$H(1,4) \leq D_0 = 4,12 \leq 20 = 0$$

$$H(2,0) \leq D_0 = 2 \leq 20 = 0$$

$$H(2,1) \leq D_0 = 2,23 \leq 20 = 0$$

$$H(2,2) \leq D_0 = 2,82 \leq 20 = 0$$

$$H(3,0) \leq D_0 = 3 \leq 20 = 0$$

$$H(3,1) \leq D_0 = 3,16 \leq 20 = 0$$

$$H(3,2) \leq D_0 = 3,6 \leq 20 = 0$$

$$H(2, 3) \leq D_0 = 3,6 \leq 20 = 0$$

$$H(3, 3) \leq D_0 = 4,24 \leq 20 = 0$$

$$H(2, 4) \leq D_0 = 4,47 \leq 20 = 0$$

$$H(3, 4) \leq D_0 = 5 \leq 20 = 0$$

$$H(4, 0) \leq D_0 = 4 \leq 20 = 0$$

$$H(4, 1) \leq D_0 = 4,12 \leq 20 = 0$$

$$H(4, 2) \leq D_0 = 4,47 \leq 20 = 0$$

$$H(4, 3) \leq D_0 = 5 \leq 20 = 0$$

$$H(4, 4) \leq D_0 = 5,65 \leq 20 = 0$$

- Kedua, *Butterworth High-Pass Filter* (BHPF) dengan menggunakan persamaan 2.13, dengan nilai $D_0 = 20$ dan $n = 2$ didapatkan nilai kernel sebagai berikut:

Butterworth High-Pass Filter (BHPF)

$$H(0, 0) = 1 - \frac{1}{1 + \left[\frac{0,0}{D_0}\right]^{2n}} = 1 - \frac{1}{1 + \left[\frac{0}{20}\right]^{2.2}} = 0$$

$$H(0, 1) = 1 - \frac{1}{1 + \left[\frac{0,1}{D_0}\right]^{2n}} = 1 - \frac{1}{1 + \left[\frac{1}{20}\right]^{2.2}} = 1 - \frac{1}{1 + 0,00000625} = 0,00000625$$

$$H(0, 2) = 1 - \frac{1}{1 + \left[\frac{2}{20}\right]^{2.2}} = 1 - \frac{1}{1,0001} = 0,0000999$$

$$H(0, 3) = 1 - \frac{1}{1 + \left[\frac{3}{20}\right]^{2.2}} = 1 - \frac{1}{1,0005062} = 0,0005059$$

$$H(0, 4) = 1 - \frac{1}{1 + \left[\frac{4}{20}\right]^{2.2}} = 1 - \frac{1}{1,0016} = 0,00159744$$

$$H(1, 0) = 1 - \frac{1}{1 + \left[\frac{1}{20}\right]^{2.2}} = 1 - \frac{1}{1,00000625} = 0,00000625$$

$$H(1, 1) = 1 - \frac{1}{1 + \left[\frac{1,41}{20}\right]^{2.2}} = 1 - \frac{1}{1,0000247} = 0,0000247$$

$$H(1, 2) = 1 - \frac{1}{1 + \left[\frac{2,23}{20}\right]^{2.2}} = 1 - \frac{1}{1,0001545} = 0,0001545$$

$$H(1, 3) = 1 - \frac{1}{1 + \left[\frac{3,16}{20}\right]^{2.2}} = 1 - \frac{1}{1,000623} = 0,0006228$$

$$H(1, 4) = 1 - \frac{1}{1 + \left[\frac{4,12}{20}\right]^{2.2}} = 1 - \frac{1}{1,0018} = 0,001797$$

$$H(2, 0) = 1 - \frac{1}{1 + \left[\frac{2}{20}\right]^{2.2}} = 1 - \frac{1}{1,0001} = 0,0000999$$

$$H(2, 1) = 1 - \frac{1}{1 + \left[\frac{2,23}{20}\right]^{2.2}} = 1 - \frac{1}{1,0001545} = 0,0001545$$

$$H(2, 2) = 1 - \frac{1}{1 + \left[\frac{2,82}{20}\right]^{2.2}} = 1 - \frac{1}{1,000395} = 0,000394$$

$$H(2, 3) = 1 - \frac{1}{1 + \left[\frac{3,6}{20}\right]^{2.2}} = 1 - \frac{1}{1,001049} = 0,001047$$

$$H(2, 4) = 1 - \frac{1}{1 + \left[\frac{4,47}{20}\right]^{2.2}} = 1 - \frac{1}{1,002495} = 0,002488$$

$$H(3, 0) = 1 - \frac{1}{1 + \left[\frac{3}{20}\right]^{2.2}} = 1 - \frac{1}{1,0005062} = 0,000505$$

$$H(3, 1) = 1 - \frac{1}{1 + \left[\frac{3,16}{20}\right]^{2.2}} = 1 - \frac{1}{1,000623} = 0,0006228$$

$$H(3, 2) = 1 - \frac{1}{1 + \left[\frac{3,6}{20}\right]^{2.2}} = 1 - \frac{1}{1,001049} = 0,001047$$

$$H(3, 3) = 1 - \frac{1}{1 + \left[\frac{4,24}{20}\right]^{2.2}} = 1 - \frac{1}{1,002019} = 0,002015$$

$$H(3, 4) = 1 - \frac{1}{1 + \left[\frac{5}{20}\right]^{2.2}} = 1 - \frac{1}{1,003906} = 0,003891$$

$$H(4, 0) = 1 - \frac{1}{1 + \left[\frac{4}{20}\right]^{2.2}} = 1 - \frac{1}{1,0016} = 0,001597$$

$$H(4, 1) = 1 - \frac{1}{1 + \left[\frac{4,12}{20}\right]^{2.2}} = 1 - \frac{1}{1,0018} = 0,001797$$

$$H(4, 2) = 1 - \frac{1}{1 + \left[\frac{4,47}{20}\right]^{2.2}} = 1 - \frac{1}{1,002495} = 0,002488$$

$$H(4, 3) = 1 - \frac{1}{1 + \left[\frac{5}{20}\right]^{2.2}} = 1 - \frac{1}{1,003906} = 0,003891$$

$$H(4, 4) = 1 - \frac{1}{1 + \left[\frac{5,65}{20}\right]^{2.2}} = 1 - \frac{1}{1,006369} = 0,006328$$

- Ketiga, *Gaussian High-Pass Filter* (GHPF) dengan menggunakan persamaan 2.14, dengan nilai $D_0 = 20$ didapatkan nilai matriks sebagai berikut:

Gaussian High-Pass Filter (GHPF)

$$H(0, 0) = 1 - (e^{-D^2(0,0)/2D_0^2}) = 1 - \left(e^{-\frac{0}{2.(20)^2}}\right) = 1 - \left(e^{-\frac{0}{800}}\right) = 1 - 1 = 0$$

$$\begin{aligned} H(0, 1) &= 1 - (e^{-D^2(0,1)/2D_0^2}) = 1 - \left(e^{-\frac{1}{2.(20)^2}}\right) = 1 - \left(e^{-\frac{1}{800}}\right) \\ &= 1 - 0,998750 \\ &= 0,00125 \end{aligned}$$

$$H(0, 2) = 1 - \left(e^{-\frac{2}{2.(20)^2}}\right) = 1 - \left(e^{-\frac{2}{800}}\right) = 1 - 0,997503 = 0,002497$$

$$H(0, 3) = 1 - \left(e^{-\frac{3}{2.(20)^2}}\right) = 1 - \left(e^{-\frac{3}{800}}\right) = 1 - 0,996257 = 0,003743$$

$$H(0, 4) = 1 - \left(e^{-\frac{4}{2.(20)^2}}\right) = 1 - \left(e^{-\frac{4}{800}}\right) = 1 - 0,995012 = 0,004988$$

$$H(1, 0) = 1 - \left(e^{-\frac{1}{2.(20)^2}}\right) = 1 - \left(e^{-\frac{1}{800}}\right) = 1 - 0,998750 = 0,00125$$

$$H(1, 1) = 1 - \left(e^{-\frac{1,41}{2.(20)^2}}\right) = 1 - \left(e^{-\frac{1,41}{800}}\right) = 1 - 0,998239 = 0,001761$$

$$H(1, 2) = 1 - \left(e^{-\frac{2,23}{2.(20)^2}}\right) = 1 - \left(e^{-\frac{2,23}{800}}\right) = 1 - 0,997216 = 0,002784$$

$$H(1, 3) = 1 - \left(e^{-\frac{3,16}{2.(20)^2}}\right) = 1 - \left(e^{-\frac{3,16}{800}}\right) = 1 - 0,996057 = 0,003943$$

$$H(1, 4) = 1 - \left(e^{-\frac{4,12}{2.(20)^2}}\right) = 1 - \left(e^{-\frac{4,12}{800}}\right) = 1 - 0,994863 = 0,005137$$

$$H(2, 0) = 1 - \left(e^{-\frac{2}{2.(20)^2}}\right) = 1 - \left(e^{-\frac{2}{800}}\right) = 1 - 0,997503 = 0,002497$$

$$H(2, 1) = 1 - \left(e^{-\frac{2,23}{2.(20)^2}}\right) = 1 - \left(e^{-\frac{2,23}{800}}\right) = 1 - 0,997216 = 0,002784$$

$$H(2, 2) = 1 - \left(e^{-\frac{2,82}{2.(20)^2}}\right) = 1 - \left(e^{-\frac{2,82}{800}}\right) = 1 - 0,996481 = 0,003519$$

$$H(2, 3) = 1 - \left(e^{-\frac{3,6}{2 \cdot (20)^2}} \right) = 1 - \left(e^{-\frac{3,6}{800}} \right) = 1 - 0,995510 = 0,00449$$

$$H(2, 4) = 1 - \left(e^{-\frac{4,47}{2 \cdot (20)^2}} \right) = 1 - \left(e^{-\frac{4,47}{800}} \right) = 1 - 0,994428 = 0,005572$$

$$H(3, 0) = 1 - \left(e^{-\frac{3}{2 \cdot (20)^2}} \right) = 1 - \left(e^{-\frac{3}{800}} \right) = 1 - 0,996257 = 0,003743$$

$$H(3, 1) = 1 - \left(e^{-\frac{3,16}{2 \cdot (20)^2}} \right) = 1 - \left(e^{-\frac{3,16}{800}} \right) = 1 - 0,996057 = 0,003943$$

$$H(3, 2) = 1 - \left(e^{-\frac{3,6}{2 \cdot (20)^2}} \right) = 1 - \left(e^{-\frac{3,6}{800}} \right) = 1 - 0,995510 = 0,00449$$

$$H(3, 3) = 1 - \left(e^{-\frac{4,24}{2 \cdot (20)^2}} \right) = 1 - \left(e^{-\frac{4,24}{800}} \right) = 1 - 0,994714 = 0,005286$$

$$H(3, 4) = 1 - \left(e^{-\frac{5}{2 \cdot (20)^2}} \right) = 1 - \left(e^{-\frac{5}{800}} \right) = 1 - 0,993769 = 0,006231$$

$$H(4, 0) = 1 - \left(e^{-\frac{4}{2 \cdot (20)^2}} \right) = 1 - \left(e^{-\frac{4}{800}} \right) = 1 - 0,995012 = 0,004988$$

$$H(4, 1) = 1 - \left(e^{-\frac{4,12}{2 \cdot (20)^2}} \right) = 1 - \left(e^{-\frac{4,12}{800}} \right) = 1 - 0,994863 = 0,005137$$

$$H(4, 2) = 1 - \left(e^{-\frac{4,47}{2 \cdot (20)^2}} \right) = 1 - \left(e^{-\frac{4,47}{800}} \right) = 1 - 0,994428 = 0,005572$$

$$H(4, 3) = 1 - \left(e^{-\frac{5}{2 \cdot (20)^2}} \right) = 1 - \left(e^{-\frac{5}{800}} \right) = 1 - 0,993769 = 0,006231$$

$$H(4, 4) = 1 - \left(e^{-\frac{5,65}{2 \cdot (20)^2}} \right) = 1 - \left(e^{-\frac{5,65}{800}} \right) = 1 - 0,992962 = 0,007038$$

- Lakukan perkalian dari matriks citra hasil transformasi fourier dengan matriks filter domain frekuensi.

Ideal High-Pass Filter

Matriks Transformasi Fourier

3341	-854,958 - 370,477 <i>i</i>	81,958 - 378,801 <i>i</i>	81,958 + 378,801 <i>i</i>	-854,958 + 370,477 <i>i</i>
-514,109 - 191,522 <i>i</i>	-125,640 + 249,042 <i>i</i>	12,458 - 37,053 <i>i</i>	-183,955 - 164,590 <i>i</i>	2,799 + 246,456 <i>i</i>
-388,890 + 69,581 <i>i</i>	70,955 + 4,737 <i>i</i>	90,140 + 32,998 <i>i</i>	139,200 - 222,264 <i>i</i>	79,541 - 156,961 <i>i</i>

$-388,890$ $- 69,581 i$	$79,541$ $+ 156,961 i$	$139,200$ $+ 222,264 i$	$90,140$ $- 32,998 i$	$70,955$ $- 4,737 i$
$-514,109$ $+ 191,522 i$	$2,799$ $- 246,456 i$	$-183,955$ $+ 164,590 i$	$12,458$ $+ 37,053 i$	$-125,640$ $- 249,042 i$

Matriks Kernel Filter IHPF

0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

Maka, matriks akhir untuk *Ideal High-Pass Filter* (IHPF) adalah:

0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

Butterworth High-Pass Filter (BHPF)

Maka, perkalian matriks antara matriks transformasi fourier dengan matriks BHPF adalah sebagai berikut:

1. $3341 * 0 = 0$
2. $(-854,958 - 370,477 i) * 0,00000625 = -0,0053434 - 0,0023154 i$
3. $(81,958 - 378,801 i) * 0,0000999 = 0,0081876 - 0,0378422 i$
4. $(81,958 + 378,801 i) * 0,0005059 = 0,0414625 + 0,1916354 i$
5. $(-854,958 + 370,477 i) * 0,00159744 = -1,3657441 + 0,5918147 i$
6. $(-514,109 - 191,522 i) * 0,00000625 = -0,0032131 - 0,001197 i$
7. $(-125,640 + 249,042 i) * 0,0000247 = -0,0031033 + 0,0061513 i$
8. $(12,458 - 37,053 i) * 0,0001545 = 0,0019247 - 0,0057246 i$
9. $(-183,955 - 164,590 i) * 0,0006228 = -0,1145671 - 0,1025066 i$
10. $(2,799 + 246,456 i) * 0,001797 = 0,0050298 + 0,4428814 i$
11. $(-388,890 + 69,581 i) * 0,0000999 = -0,0388501 + 0,0069511 i$

12. $(70,955 + 4,737 i) * 0,0001545 = 0,0109625 + 0,0007318 i$
13. $(90,140 + 32,998 i) * 0,000394 = 0,0355151 + 0,0130012 i$
14. $(139,200 - 222,264 i) * 0,001047 = 0,1457424 - 0,2327104 i$
15. $(79,541 - 156,961 i) * 0,002488 = 0,197898 - 0,3905189 i$
16. $(-388,890 - 69,581 i) * 0,000505 = -0,1963894 - 0,035138 i$
17. $(79,541 + 156,961 i) * 0,0006228 = 0,049538 + 0,097755 i$
18. $(139,200 + 222,264 i) * 0,001047 = 0,145742 + 0,232710 i$
19. $(90,140 - 32,998 i) * 0,002015 = 0,181632 - 0,066490 i$
20. $(70,955 - 4,737 i) * 0,003891 = 0,276085 - 0,018431 i$
21. $(-514,109 + 191,522 i) * 0,001597 = -0,821237 + 0,305937 i$
22. $(2,799 - 246,456 i) * 0,001797 = 0,005029 - 0,442881 i$
23. $(-183,955 + 164,590 i) * 0,002488 = -0,457680 + 0,409499 i$
24. $(12,458 + 37,053 i) * 0,003891 = 0,048474 + 0,144173 i$
25. $(-125,640 - 249,042 i) * 0,006328 = -0,795049 - 1,575937 i$

Maka, nilai matriks hasil perkalian dari matriks transformasi fourier dan *Butterworth High-Pass Filter* (BHPF) adalah:

0	-0,0053434 - 0,0023154 i	0,0081876 - 0,0378422 i	0,0414625 +0,1916354 i	-1,3657441 + 0,5918147 i
-0,0032131 - 0,001197 i	-0,0031033 + 0,0061513 i	0,0019247 - 0,0057246 i	-0,1145671 - 0,1025066 i	0,0050298 + 0,4428814 i
-0,0388501 + 0,0069511 i	0,0109625 + 0,0007318 i	0,0355151 + 0,0130012 i	0,1457424 - 0,2327104 i	0,197898 - 0,3905189 i
-0,1963894 - 0,035138 i	0,049538 + 0,097755 i	0,145742 + 0,232710 i	0,181632 - 0,066490 i	0,276085 - 0,018431 i
-0,821237 + 0,305937 i	0,005029 - 0,442881 i	-0,457680 + 0,409499 i	0,048474 + 0,144173 i	-0,795049 - 1,575937 i

Gaussian High-Pass Filter (GHPF)

Maka, perkalian matriks antara matriks transformasi fourier dengan matriks GHPF adalah sebagai berikut:

1. $3341 * 0 = 0$
2. $(-854,958 - 370,477 i) * 0,00125 = -1,068697 - 0,463096 i$
3. $(81,958 - 378,801 i) * 0,002497 = 0,204649 - 0,945866 i$
4. $(81,958 + 378,801 i) * 0,003743 = 0,306768 + 1,417852 i$
5. $(-854,958 + 370,477 i) * 0,004988 = -4,26453 + 1,847939 i$
6. $(-514,109 - 191,522 i) * 0,00125 = -0,642636 - 0,239402 i$
7. $(-125,640 + 249,042 i) * 0,001761 = -0,221252 + 0,438562 i$
8. $(12,458 - 37,053 i) * 0,002784 = 0,034683 - 0,103155 i$
9. $(-183,955 - 164,590 i) * 0,003943 = -0,725334 - 0,648978 i$
10. $(2,799 + 246,456 i) * 0,005137 = 0,014378 + 1,266044 i$
11. $(-388,890 + 69,581 i) * 0,002497 = -0,971058 + 0,173743 i$
12. $(70,955 + 4,737 i) * 0,002784 = 0,197538 + 0,013187 i$
13. $(90,140 + 32,998 i) * 0,003519 = 0,317202 + 0,116119 i$
14. $(139,200 - 222,264 i) * 0,00449 = 0,625008 - 0,997965 i$
15. $(79,541 - 156,961 i) * 0,005572 = 0,443202 - 0,874586 i$
16. $(-388,890 - 69,581 i) * 0,003743 = -1,455615 - 0,260441 i$
17. $(79,541 + 156,961 i) * 0,003943 = 0,313630 + 0,618897 i$
18. $(139,200 + 222,264 i) * 0,00449 = 0,625008 + 0,997965 i$
19. $(90,140 - 32,998 i) * 0,005286 = 0,476480 - 0,174427 i$
20. $(70,955 - 4,737 i) * 0,006231 = 0,442120 - 0,029516 i$
21. $(-514,109 + 191,522 i) * 0,004988 = -2,564375 + 0,955311 i$
22. $(2,799 - 246,456 i) * 0,005137 = 0,014378 - 1,266044 i$
23. $(-183,955 + 164,590 i) * 0,005572 = -1,024997 + 0,917095 i$
24. $(12,458 + 37,053 i) * 0,006231 = 0,077625 + 0,230877 i$
25. $(-125,640 - 249,042 i) * 0,007038 = -0,884254 - 1,752757 i$

Maka, nilai matriks hasil perkalian dari matriks transformasi fourier dan *Gaussian High-Pass Filter* (GHPF) adalah:

0	-1,068697 - 0,463096 <i>i</i>	0,204649 - 0,945866 <i>i</i>	0,306768 + 1,417852 <i>i</i>	-4,26453 + 1,847939 <i>i</i>
-0,642636 - 0,239402 <i>i</i>	-0,221252 + 0,438562 <i>i</i>	0,034683 - 0,103155 <i>i</i>	-0,725334 - 0,648978 <i>i</i>	0,014378 + 1,266044 <i>i</i>
-0,971058 + 0,173743 <i>i</i>	0,197538 + 0,013187 <i>i</i>	0,317202 + 0,116119 <i>i</i>	0,625008 - 0,997965 <i>i</i>	0,443202 - 0,874586 <i>i</i>
-1,455615 - 0,260441 <i>i</i>	0,313630 + 0,618897 <i>i</i>	0,625008 + 0,997965 <i>i</i>	0,476480 - 0,174427 <i>i</i>	0,442120 - 0,029516 <i>i</i>
-2,564375 + 0,955311 <i>i</i>	0,014378 - 1,266044 <i>i</i>	-1,024997 + 0,917095 <i>i</i>	0,077625 + 0,230877 <i>i</i>	-0,884254 - 1,752757 <i>i</i>

3. Invers Fourier Transform

$$F(x, y) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(u, v) \left(\cos \left(2\pi \left(\frac{ux}{M} + \frac{vy}{N} \right) \right) + j \sin \left(2\pi \left(\frac{ux}{M} + \frac{vy}{N} \right) \right) \right) \dots (4.6)$$

Dimana:

M = baris

N = kolom

u = frekuensi bernilai dari 0 sampai dengan M-1

v = frekuensi bernilai dari 0 sampai dengan N-1

x = koordinat baris piksel

y = koordinat kolom piksel

π = 180

$F(u, v)$ = koefisien fourier

Maka, perhitungan manual untuk *invers* dari *filter* GHPF adalah sebagai berikut:

$$\begin{aligned}
 F(0, 0) &= \frac{1}{25} (0[1 - 0] + (-1,068697)[1 - 0] + 0,204649[1 - 0] \\
 &\quad + 0,306768[1 - 0] \\
 &\quad + (-4,26453)[1 - 0] + (-0,642636)[1 - 0] + (-0,221252)[1 - 0] \\
 &\quad + 0,034683[1 - 0] + (-0,725334)[1 - 0] + 0,014378[1 - 0]
 \end{aligned}$$

$$\begin{aligned}
& +(-0,971058)[1-0] + 0,197538[1-0] + 0,317202[1-0] \\
& \quad + 0,625008[1-0] \\
& +0,443202[1-0] + (-1,455615)[1-0] + 0,313630[1-0] \\
& \quad + 0,625008[1-0] \\
& +0,476480[1-0] + 0,442120[1-0] + -2,564375[1-0] \\
& \quad + 0,014378[1-0] \\
& +(-1,024997)[1-0] + 0,077625[1-0] + -0,884254[1-0] \\
\mathbf{F(0,0)} &= \frac{1}{25}([0-0] + [-1,068697-0] + [0,204649-0] + [0,306768-0] \\
& +[-4,26453-0] + [-0,642636-0] + [-0,221252-0] + [0,034683-0] \\
& +[-0,725334-0] + [0,014378-0] + [-0,971058-0] + [0,197538-0] \\
& +[0,317202-0] + [0,625008-0] + [0,443202-0] + [-1,455615-0] \\
& +[0,313630-0] + [0,625008-0] + [0,476480-0] + [0,442120-0] \\
& +[-2,564375-0] + [0,014378-0] + [-1,024997-0] + [0,077625-0] \\
& +[-0,884254-0] \\
\mathbf{F(0,0)} &= (2,143705 + 0\ i) \\
\mathbf{F(0,1)} &= \frac{1}{25}(0[1-0] + (-1,068697)[0,309-0,951\ i] \\
& \quad + 0,204649[-0,809-0,587\ i] \\
& +0,306768[-0,809-(-0,587)i] + (-4,26453)[0,309-(-0,951)i] + \\
& \quad \dots\dots\dots \\
& \pm 2,564375[1-0] + 0,014378[0,309-0,951\ i] \\
& \quad + (-1,024997)[-0,809-0,587\ i] \\
& +0,077625[-0,809-(-0,587)\ i] + -0,884254[0,309-(-0,951)\ i] \\
\mathbf{F(0,1)} &= (-2,66756 - 1,155938\ i) \\
\mathbf{F(0,2)} &= \frac{1}{25}(0(1-0) \pm 1,068697((-0,809) - 0,587\ i) \\
& +0,204649(0,309 - (-0,951)i) + 0,306768(0,309 - 0,951\ i) \\
& +(-4,26453)(-0,809) - (-0,587)\ i) + \dots\dots\dots \\
& + -2,564375(1-0) + 0,014378((-0,809) - 0,587\ i) \\
& + -1,024997(0,309 - (-0,951)i)
\end{aligned}$$

$$+0,077625(0,309 - 0,951 i) + -0,884254(-0,809) - (-0,587)i$$

$$F(0,2) = (0,663196 - 3,065293 i)$$

$$F(0,3) = \frac{1}{25}(0(1-0) \pm 1,068697((-0,809) - 0,587 i)$$

$$+0,204649(0,309 - 0,951)i + 0,306768(0,309 - (-0,951)i)$$

$$+(-4,26453)(-0,809) - 0,587 i) + \dots$$

$$+ -2,564375(1-0) + 0,014378((-0,809) - 0,587 i)$$

$$+ -1,024997(0,309 - (-0,951)i)$$

$$+0,077625(0,309 - (-0,951)i) + -0,884254(-0,809) - 0,587 i)$$

$$F(0,3) = (0,255713 + 1,181907 i)$$

$$F(0,4) = \frac{1}{25}(0(1-0) \pm 1,068697(0,309 - (-0,951) i)$$

$$+0,204649(-0,809 - (-0,587i)) + 0,306768(-0,809 - 0,587i)$$

$$+(-4,26453)(0,309 - 0,951 i) + \dots$$

$$+ -2,564375(1-0) + 0,014378(0,309 - (-0,951)i)$$

$$+ -1,024997(-0,809 - (-0,587i))$$

$$+0,077625(-0,809 - 0,587i) + -0,884254(0,309 - 0,951)i)$$

$$F(0,4) = (-0,534180 + 0,231476 i)$$

$$F(1,0) = (-1,604085 - 0,597574 i)$$

$$F(1,1) = (-0,704742 + 1,396933 i)$$

$$F(1,2) = (0,131675 - 0,391611 i)$$

$$F(1,3) = (-1,031847 - 0,923222 i)$$

$$F(1,4) = (0,008736 + 0,768974 i)$$

$$F(2,0) = (-3,14693 + 0,563057 i)$$

$$F(2,1) = (0,749915 + 0,050074 i)$$

$$F(2,2) = (1,397495 + 0,511598 i)$$

$$F(2,3) = (1,471171 - 2,349063 i)$$

$$F(2,4) = (0,643652 - 1,270148 i)$$

$$F(3,0) = (-1,213384 - 0,217102 i)$$

$$F(3,1) = (0,446162 + 0,880431 i)$$

$$F(3,2) = (1,471171 + 2,349063 i)$$

$$F(3, 3) = (0,505615 - 0,185097 i)$$

$$F(3, 4) = (0,221391 - 0,014783 i)$$

$$F(4, 0) = (-0,321218 + 0,119664 i)$$

$$F(4, 1) = (0,008736 - 0,768974 i)$$

$$F(4, 2) = (-1,488585 + 1,331880 i)$$

$$F(4, 3) = (0,038873 + 0,115611 i)$$

$$F(4, 4) = (0,078500 - 0,155603 i)$$

Berikut ini merupakan matriks hasil *invers* transformasi fourier dengan filter GHPF:

(2,143705 + 0)	(-2,66756 - 1,155938 i)	(0,663196 - 3,065293 i)	(0,255713 + 1,181907 i)	(-0,534180 + 0,231476 i)
(-1,604085 - 0,597574 i)	(-0,704742 + 1,396933 i)	(0,131675 - 0,391611 i)	(-1,031847 - 0,923222 i)	(0,008736 + 0,768974 i)
(-3,14693 + 0,563057 i)	(0,749915 + 0,050074 i)	(1,397495 + 0,511598 i)	(1,471171 - 2,349063 i)	(0,643652 - 1,270148 i)
(-1,213384 - 0,217102 i)	(0,446162 + 0,880431 i)	(1,471171 + 2,349063 i)	(0,505615 - 0,185097 i)	(0,221391 - 0,014783 i)
(-0,321218 + 0,119664 i)	(0,008736 - 0,768974 i)	(-1,488585 + 1,331880 i)	(0,038873 + 0,115611 i)	(0,078500 - 0,155603 i)

g. *Mean Square Error* (MSE) dan *Peak Signal to Noise Ratio* (PSNR)

Setelah didapat citra dari hasil HPF dan FFT, langkah selanjutnya adalah menghitung nilai MSE dan PSNR. Misalnya matriks yang akan dihitung adalah matriks dari histogram ekualisasi sebagai tahap *pre-processing* dengan citra HPF sebagai hasil *process*. Dengan menggunakan persamaan 2.16 dan 2.17, dimana $f(x, y)$ adalah matriks histogram ekualisasi dan $g(x, y)$ sebagai matriks HPF. Maka perhitungannya adalah sebagai berikut:

Mean Square Error (MSE)

$$f(x, y) = \begin{bmatrix} 10 & 51 & 132 & 112 & 20 \\ 61 & 183 & 255 & 255 & 51 \\ 102 & 173 & 255 & 255 & 91 \\ 153 & 163 & 142 & 204 & 71 \\ 51 & 214 & 132 & 204 & 91 \end{bmatrix} \quad g(x, y) = \begin{bmatrix} 0 & 0 & 242 & 153 & 0 \\ 10 & 255 & 255 & 255 & 0 \\ 123 & 162 & 255 & 255 & 78 \\ 255 & 133 & 0 & 255 & 0 \\ 0 & 255 & 100 & 255 & 180 \end{bmatrix}$$

$$MSE = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} |(f(x, y) - g(x, y))|^2$$

$$MSE = \frac{1}{25} [(10 - 0)^2 + (51 - 0)^2 + (132 - 242)^2 + (112 - 153)^2 + (20 - 0)^2 + (61 - 10)^2 + (183 - 255)^2 + (255 - 255)^2 + (255 - 255)^2 + (51 - 0)^2 + (102 - 123)^2 + (173 - 162)^2 + (255 - 255)^2 + (255 - 255)^2 + (91 - 78)^2 +$$

$$(153 - 255)^2 + (163 - 133)^2 + (142 - 0)^2 + (204 - 255)^2 + (71 - 0)^2 + (51 - 0)^2 + (214 - 255)^2 + (132 - 100)^2 + (204 - 255)^2 + (91 - 180)^2]$$

$$MSE = \frac{1}{25} [100 + 2.601 + 12.100 + 1.681 + 400 + 2.601 + 5.184 + 0 + 0 + 2.601 + 441 + 121 + 0 + 0 + 169 + 10.404 + 900 + 20.164 + 2.601 + 5.041$$

$$+ 2.601 + 1.681 + 1.024 + 2.601 + 7.921]$$

$$MSE = \frac{1}{25} * 82.937$$

$$MSE = 3.31748$$

Peak Signal to Noise Ratio (PSNR)

$$PSNR = 10 \cdot \log_{10} \left(\frac{MAX^2}{MSE} \right) = 10 \cdot \log_{10} \left(\frac{255^2}{3.31748} \right) = 42.92272$$

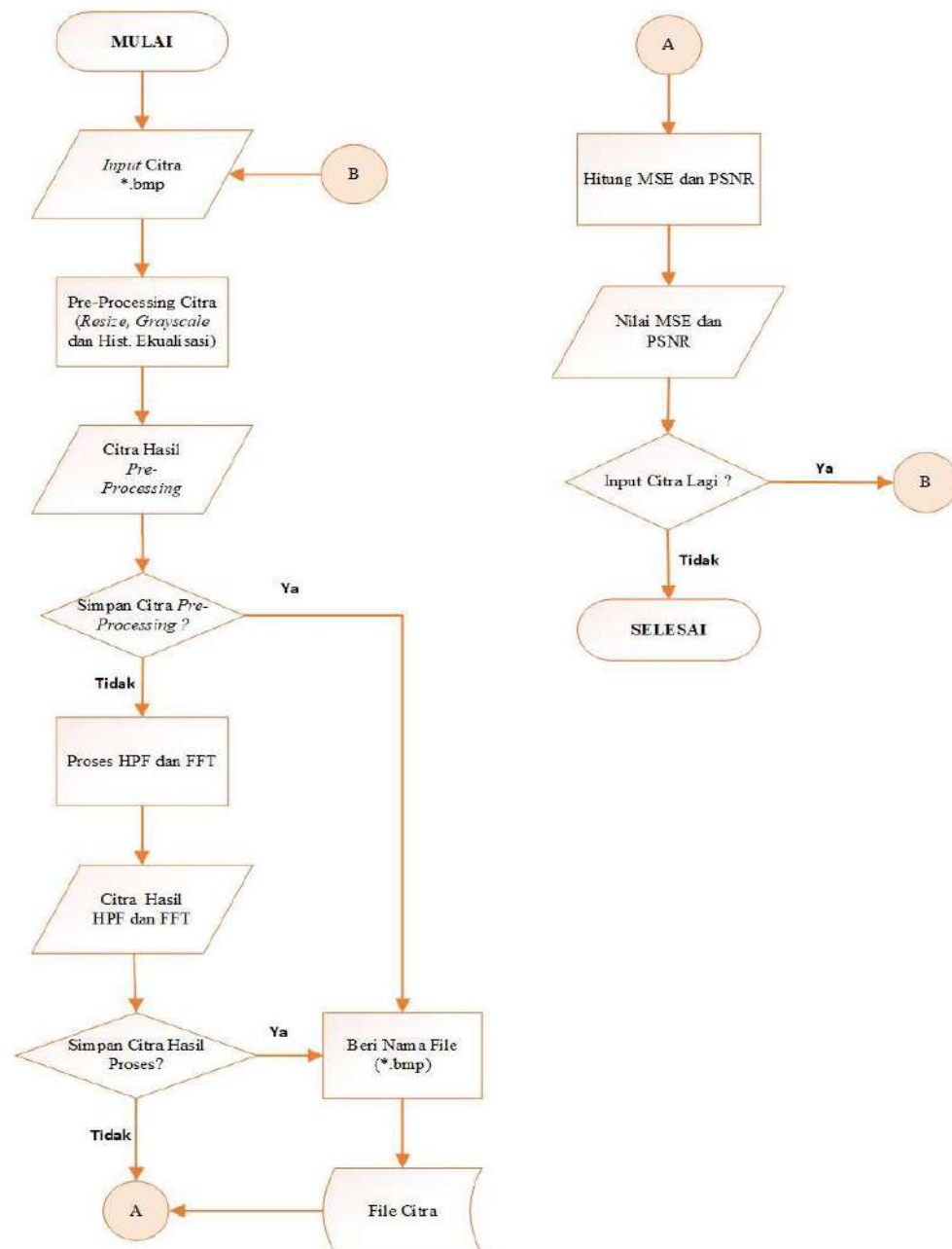
4.1.3. Perancangan Sistem

Tahap perancangan adalah tahapan setelah proses analisis kebutuhan. Perancangan pada penelitian ini dapat dilihat pada gambar 3.2 yang merupakan rancangan diagram sistem serta gambar 3.3 dan 3.4 yang merupakan rancangan *flowchart* dari sistem yang akan digunakan. Pada bab ini, perancangan akan

membahas *flowchart* sistem aplikasi dan tampilan utama program atau *user interface* dari sistem yang dibuat. Berikut ini merupakan rancangan sistem yang akan dibuat:

1. *Flowchart*

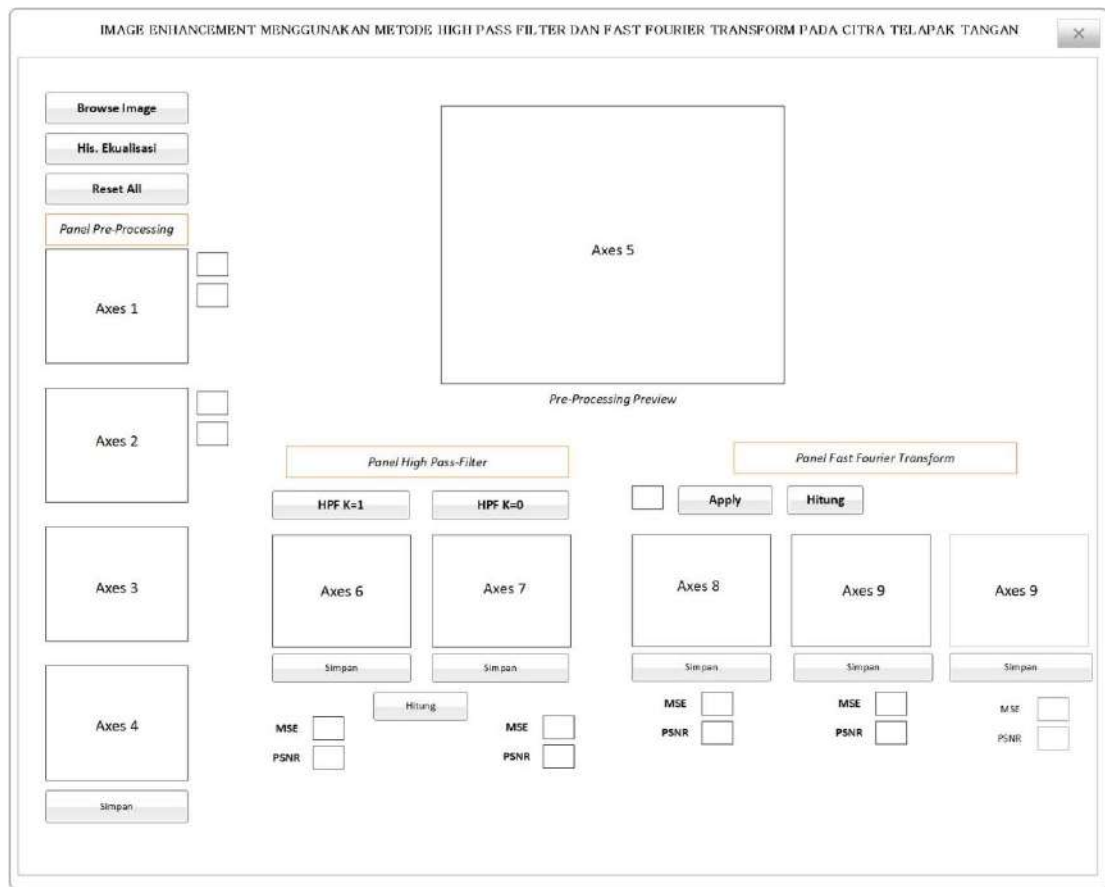
Merupakan diagram atau gambar yang menunjukkan aliran interaksi dan hubungan suatu program. Diagram alir diharapkan dapat memperjelas perkembangan program yang dibuat dalam struktur grafis sehingga orang lain dapat memahami aliran yang telah dibuat. Di bawah ini akan ditampilkan diagram alur kerangka yang telah dibuat yang dapat dilihat pada gambar yang menyertai ini:



Gambar: 4.6. Flowchart Aplikasi Image Enhancement

2. User Interface

Merupakan bentuk tampilan ilustrasi dari suatu sistem yang akan dirancang yang berhubungan langsung dengan klien. Pada penelitian ini, penulis menggunakan aplikasi Ms. Visio Pro 2016 dalam merancang *user interface* sistem yang akan dihasilkan dan dapat dilihat seperti gambar berikut:



Gambar: 4.7. Rancangan *User Interface* Aplikasi *Image Enhancement*


4.2. Hasil




Pada bidang ini, akan dijelaskan beberapa tahapan-tahapan yang diperoleh dari penelitian ini seperti tahapan pengujian, hasil data uji dan implementasi/penerapan yang akan dijelaskan seperti berikut ini:





4.2.1. Pengujian Sistem

Setelah perancangan sistem dibuat, tahapan berikutnya adalah dengan menrapkan pengujian terhadap sistem tersebut. Hal ini dikerjakan agar dapat mengenali kesalahan-kesalahan yang ada pada saat menjalankan sebuah sistem yang di bangun, sehingga dapat langsung di perbaiki dan ditingkatkan lagi. Salah satu proses pengujian yang dilakukan dalam penelitian ini adalah dengan teknik *black box testing*. Adapun pengujian *black box* yang dilakukan pada sistem ini dapat dilihat pada tabel dibawah sebagai berikut:

Tabel 4.4. Hasil Pengujian *Black Box* Pada Sistem



No.	Tombol	Fungsi	Skenario	Hasil	Ket.
1		Mengambil citra yang akan diolah	<i>User</i> menekan tombol <i>browse</i>	<ul style="list-style-type: none"> • Menampilkan kotak dialog untuk memilih citra uji • Hanya menampilkan citra berformat *.bmp • Citra uji akan langsung di <i>resize</i> dan di <i>grayscale</i> serta menampilkan histogram pada axes • Menampilkan resolusi citra sebelum dan 	Sesuai

				sesudah di <i>resize</i>	
2		Meningkatkan kontras citra menggunakan histogram ekualisasi	<i>User</i> menekan tombol Histogram ekualisasi	<ul style="list-style-type: none"> •Menampilkan histogram dari citra yang sudah di proses •Menampilkan citra hasil histogram ekualisasi, serta •Menampilkan citra pada <i>axes preview</i>. 	Sesuai
3		Membersihkan semua proses/ <i>reset</i> seperti semula.	<i>User</i> menekan tombol <i>Reset All</i>	<ul style="list-style-type: none"> •Mereset semua proses kembali seperti semula 	Sesuai
4		Menyimpan citra hasil histogram ekualisasi	<i>User</i> menekan tombol Simpan	<ul style="list-style-type: none"> •Menampilkan kotak dialog menyimpan citra •Citra yang akan disimpan dalam format *.bmp •Citra tersimpan di folder penyimpanan 	Sesuai

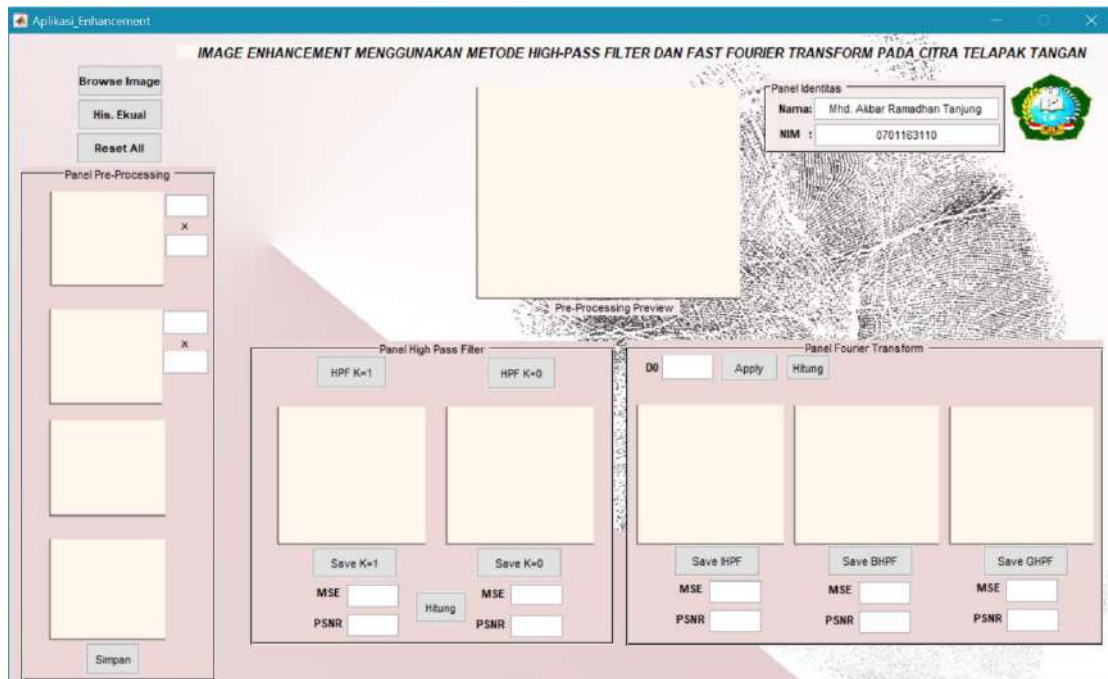
5		Melakukan penajaman terhadap citra dengan filter HPF dimana $K=1$	<i>User</i> menekan tombol HPF $K=1$	<ul style="list-style-type: none"> • Menampilkan citra yang lebih tajam dari hasil konvolusi dengan kernel filter HPF $K=1$ 	Sesuai
6		Melakukan penajaman terhadap citra dengan filter HPF dimana $K=1$	<i>User</i> menekan tombol HPF $K=0$	<ul style="list-style-type: none"> • Menampilkan citra yang lebih tajam dari hasil konvolusi dengan kernel filter HPF $K=0$ 	Sesuai
7		Menyimpan citra hasil penajaman dengan filter HPF $K=1$	<i>User</i> menekan tombol simpan	<ul style="list-style-type: none"> • Menampilkan kotak dialog menyimpan citra • Citra yang akan disimpan dalam format *.bmp • Citra tersimpan di folder penyimpanan 	Sesuai
8		Menyimpan citra hasil penajaman	<i>User</i> menekan	<ul style="list-style-type: none"> • Menampilkan kotak dialog 	Sesuai

		dengan filter HPF $K=0$	tombol simpan	menyimpan citra <ul style="list-style-type: none"> • Citra yang akan disimpan dalam format *.bmp • Citra tersimpan di folder penyimpanan 	
9	Hitung	Menghitung nilai MSE dan PSNR dari citra HPF	User menekan tombol hitung	<ul style="list-style-type: none"> • Menunjukkan hasil nilai MSE dan PSNR dari citra HPF $K=1$ • Menunjukkan nilai MSE dan PSNR dari citra HPF $K=0$ 	Sesuai
10	Apply FFT	Melakukan penajaman citra dengan filter FFT	<ul style="list-style-type: none"> • User memasukkan nilai <i>cutoff</i> • User menekan tombol <i>Apply FFT</i> 	<ul style="list-style-type: none"> • Menampilkan citra hasil transformasi fourier, spectrum center, log transform dan konvolusi pada figure • Menampilkan citra hasil 	Sesuai

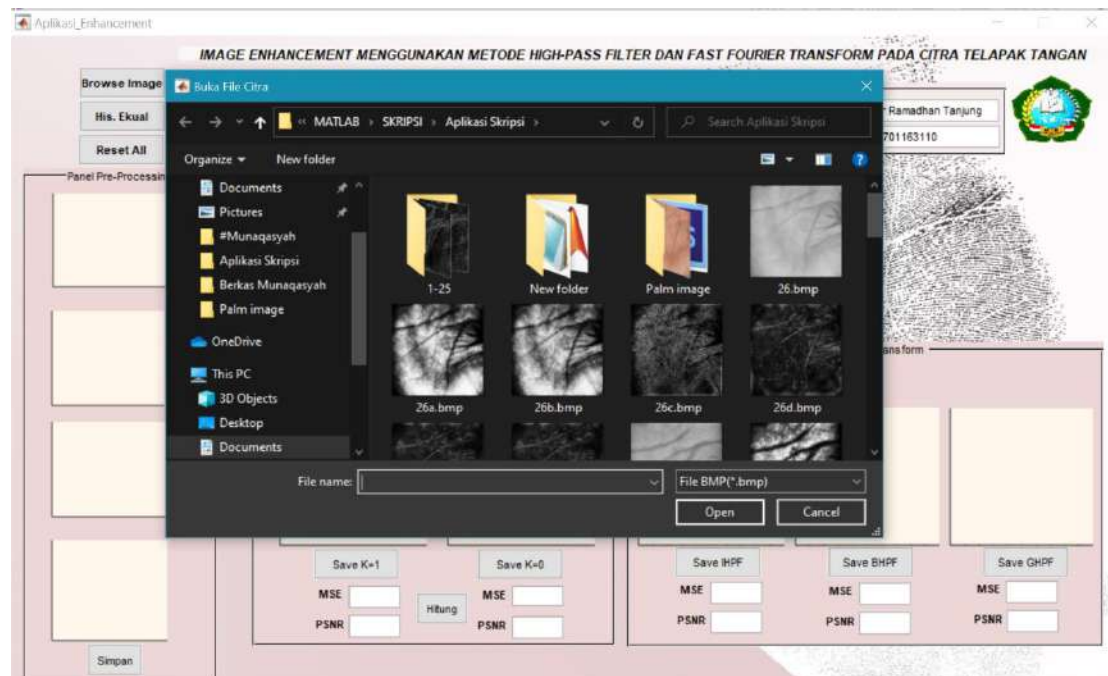
				konvolusi filter IHPF, BHPF, dan GHPF kedalam axes.	
11	Hitung	Menghitung nilai MSE dan PSNR dari citra FFT <i>filter</i>	<i>User</i> menekan tombol hitung	<ul style="list-style-type: none"> • Menampakkan hasil perhitungan MSE dan PSNR dari citra filter IHPF • Menampakkan nilai MSE dan PSNR dari citra filter BHPF • Menunjukkan nilai hasil perhitungan MSE dan PSNR dari citra filter GHPF 	Sesuai
12	Simpan	Menyimpan citra hasil penajaman dengan filter IHPF	<i>User</i> menekan tombol simpan	<ul style="list-style-type: none"> • Menampilkan kotak dialog menyimpan citra • Citra yang akan disimpan 	Sesuai

				<p>dalam format *.bmp</p> <ul style="list-style-type: none"> •Citra tersimpan di folder penyimpanan 	
13		Menyimpan citra hasil penajaman dengan filter BHPF	<i>User</i> menekan tombol simpan	<ul style="list-style-type: none"> •Menampilkan kotak dialog menyimpan citra •Citra yang akan disimpan dalam format *.bmp •Citra tersimpan di folder penyimpanan 	Sesuai
14		Menyimpan citra hasil penajaman dengan filter GHPF	<i>User</i> menekan tombol simpan	<ul style="list-style-type: none"> •Menampilkan kotak dialog menyimpan citra •Citra yang akan disimpan dalam format *.bmp •Citra tersimpan di folder penyimpanan 	Sesuai

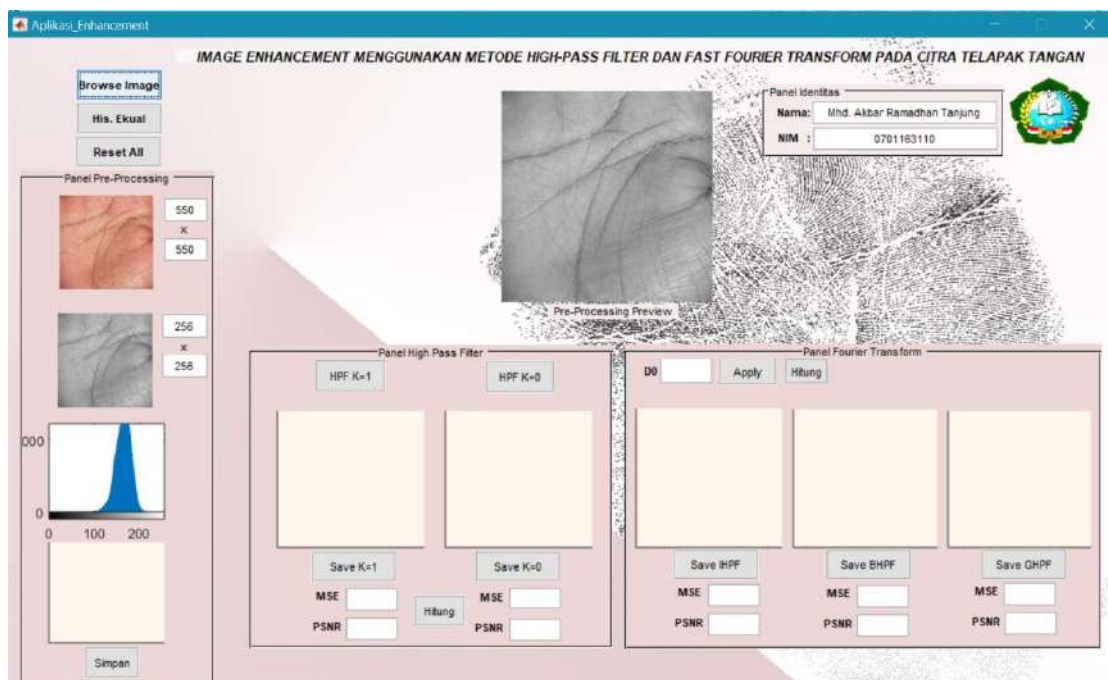
Berdasarkan hasil pengujian *black box* diatas, berikut ini merupakan tampilan *user interface* dari sistem yang telah di lakukan pengujian:



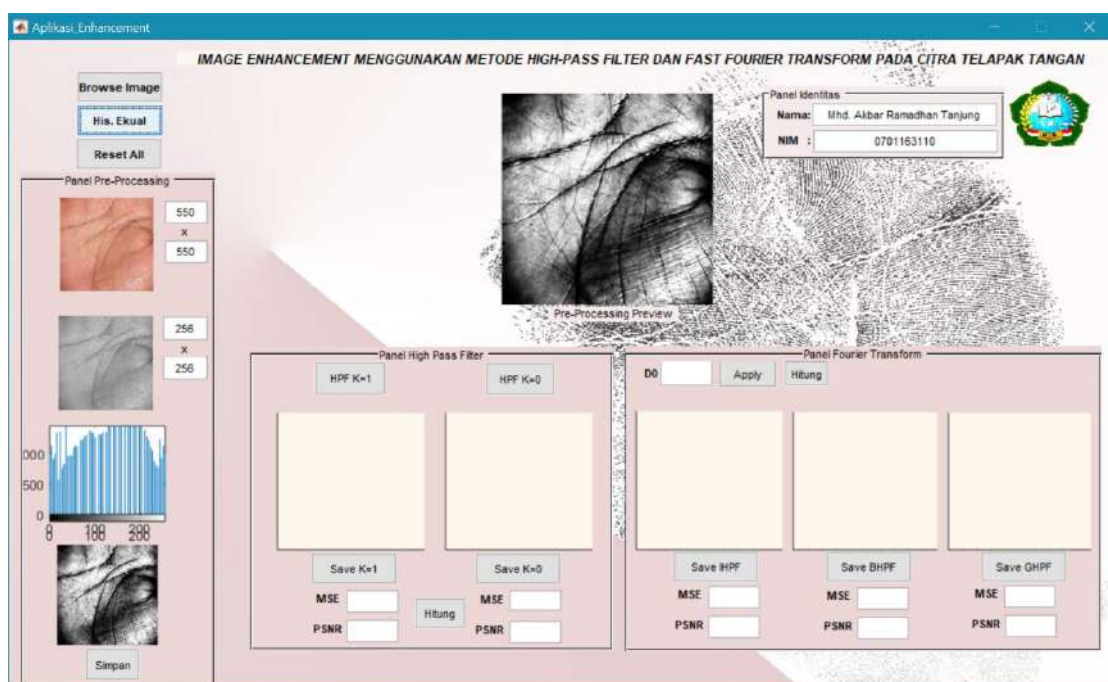
Gambar: 4.8. Tampilan Aplikasi *Image Enhancement*



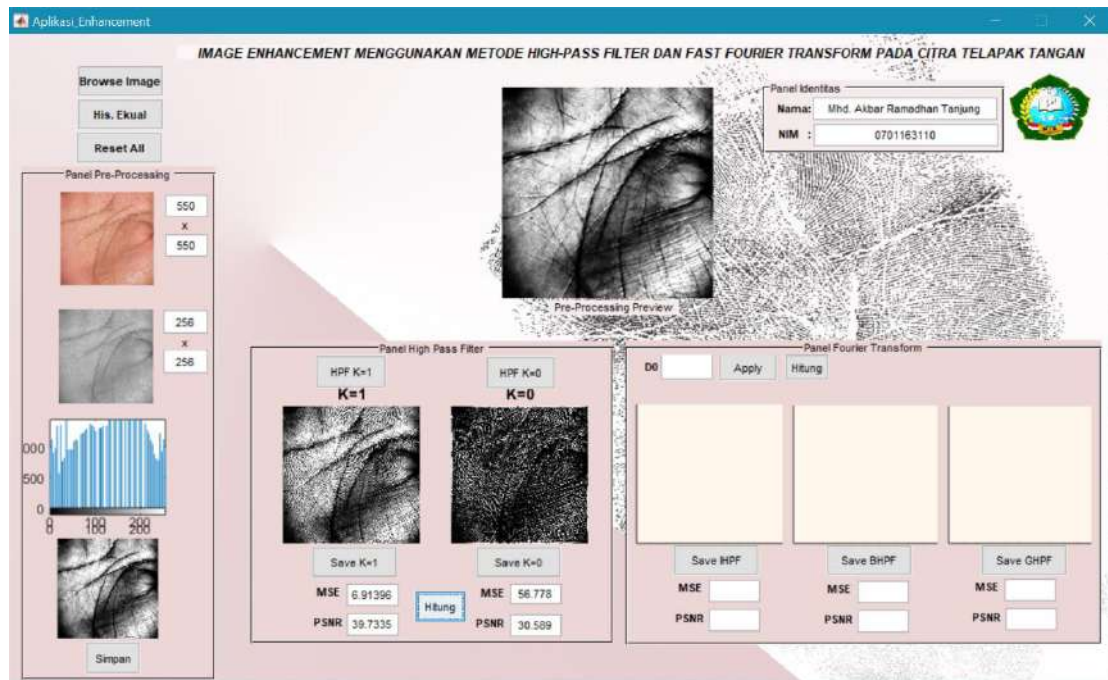
Gambar: 4.9. Tampilan “Buka File Citra”



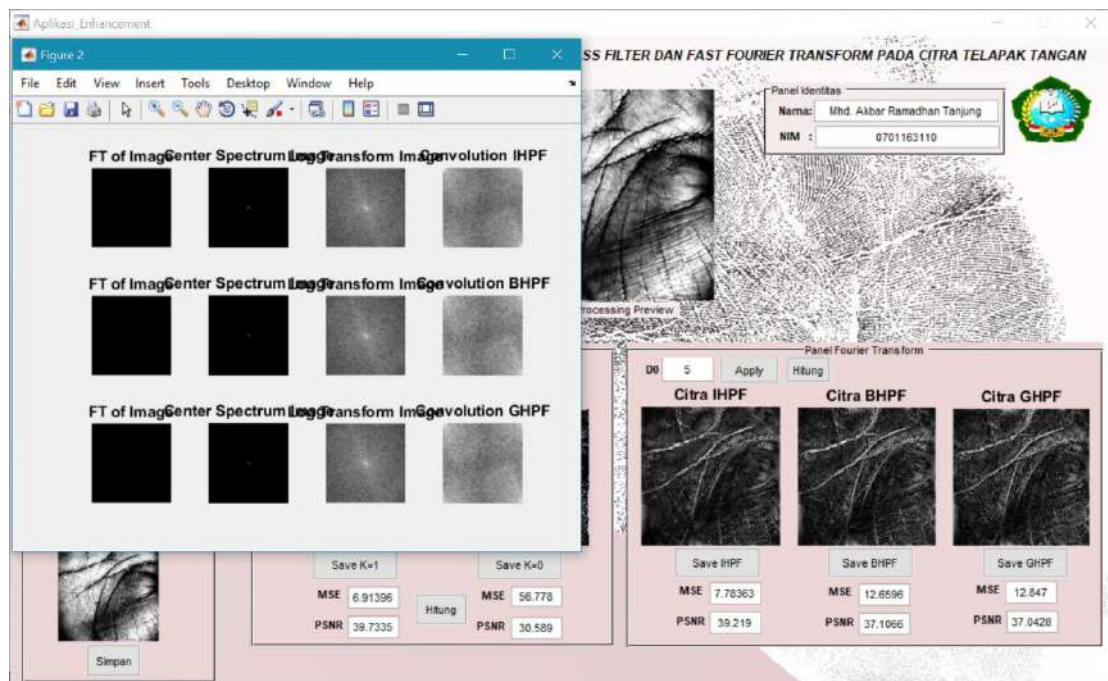
Gambar: 4.10. Tampilan Citra Setelah di Input ke Sistem



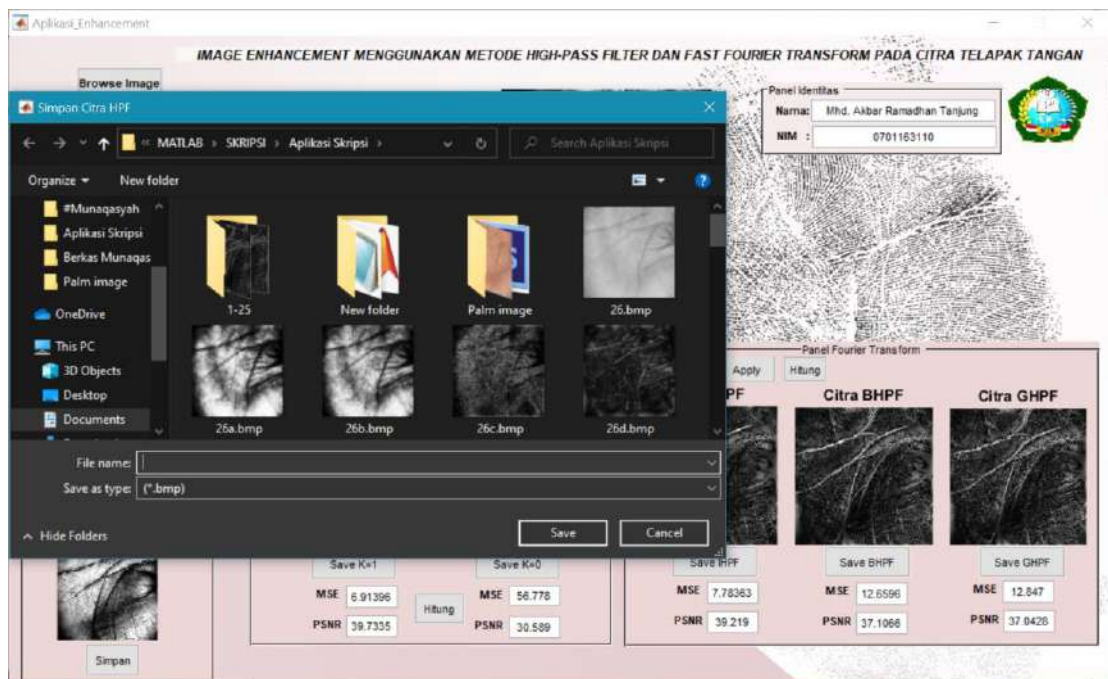
Gambar: 4.11. Tampilan Hasil Histogram Ekualisasi



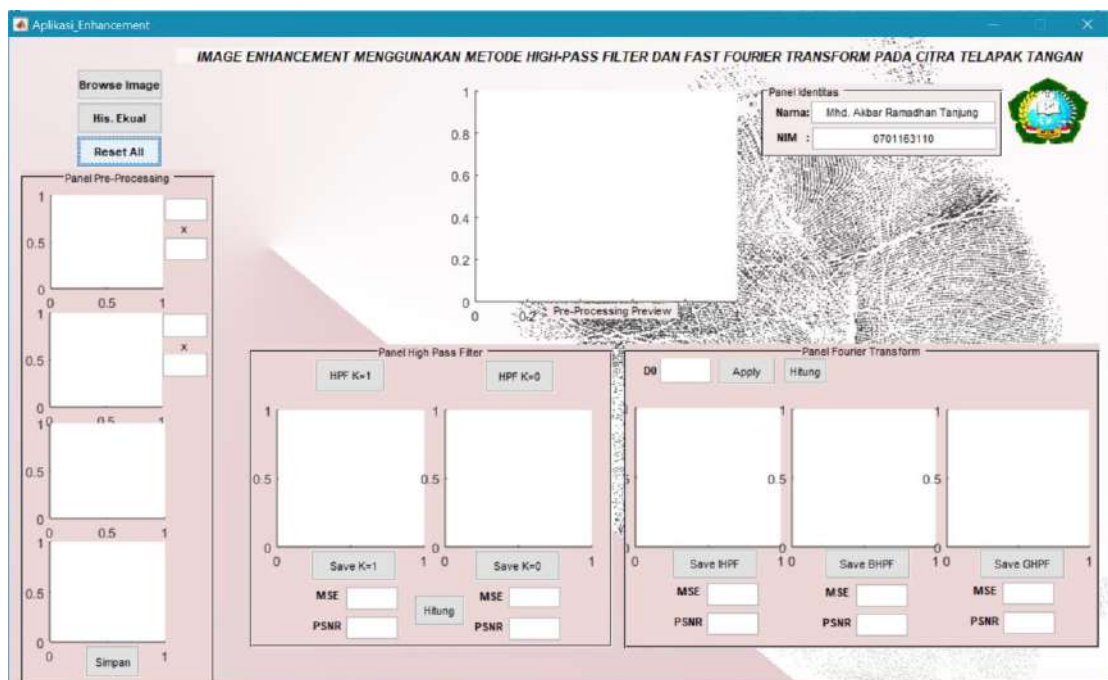
Gambar: 4.12. Tampilan Filter HPF Beserta Nilai MSE dan PSNR



Gambar: 4.13. Tampilan Citra Filter FFT, Figure Fourier Transform dan Nilai MSE, PSNR



4.14. Tampilan Simpan Citra



4.15. Tampilan Setelah di "Reset All"

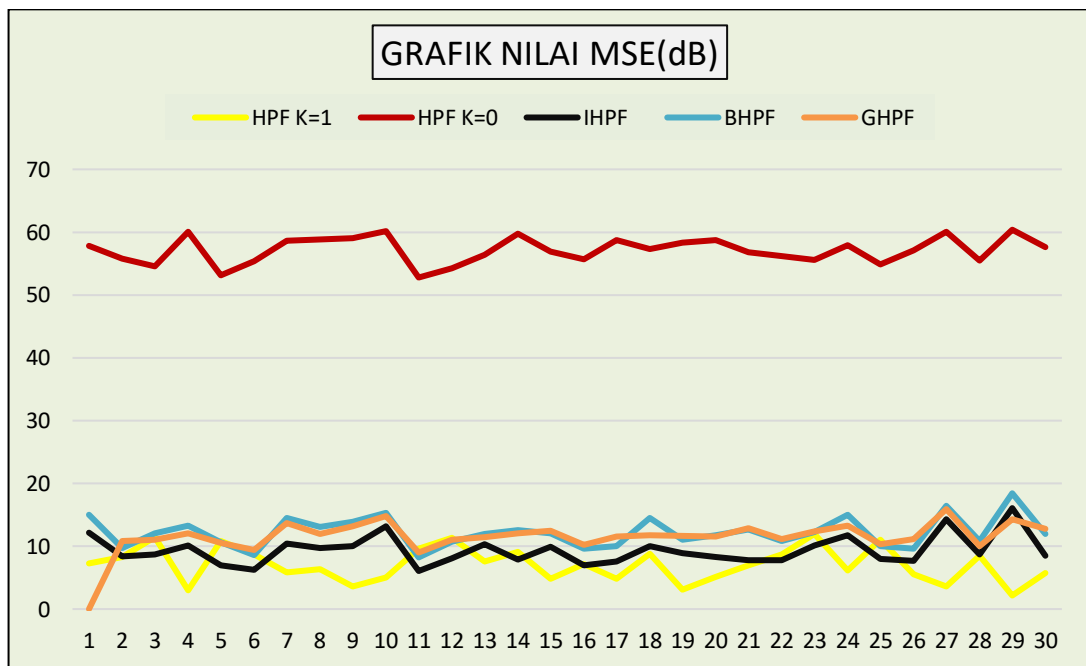
4.2.2. Hasil Data Uji

Mengingat pengujian yang telah dilakukan terhadap 30 gambar telapak tangan, di bawah ini akan ditampilkan nilai MSE dan PSNR yang telah diakuisisi dan dapat dilihat pada tabel dan grafik terlampir.:

Tabel 4.5. Nilai MSE(dB)

No	High Pass Filter		Fast Fourier Transform ($D_0=5$)		
	<i>HPF K=1</i>	<i>HPF K=0</i>	<i>Ideal HPF</i>	<i>Butterworth HPF (n=2)</i>	<i>Gaussian HPF</i>
1	7.30426	57.8302	12.19	15.049	14.9888
2	8.26257	55.7901	8.36899	9.7305	10.8839
3	11.4777	54.5919	8.67941	12.0503	11.0597
4	3.00615	60.0961	10.1678	13.3005	12.0459
5	10.8221	53.1591	6.91971	10.627	10.5737
6	8.69212	55.3829	6.28616	8.61241	9.42442
7	5.88162	58.6221	10.4411	14.5509	13.6562
8	6.35899	58.8148	9.70816	13.0912	11.9267
9	3.56204	59.0838	10.0444	13.8586	13.142
10	5.05219	60.1981	13.1627	15.331	14.7685
11	9.618	52.7791	6.05974	8.22427	9.03389
12	11.3526	54.2638	8.07271	10.7687	11.0349
13	7.57091	56.3419	10.3507	12.0061	11.4938
14	9.11024	59.7959	7.82846	12.6172	12.0105
15	4.78052	56.9404	9.88989	12.0755	12.4952
16	7.17819	55.6825	6.94823	9.60593	10.2451
17	4.82753	58.773	7.58618	9.96953	11.58
18	8.79881	57.3014	10.0261	14.5499	11.7843
19	3.05643	58.2964	8.93442	11.0856	11.6241
20	5.10147	58.7218	8.27405	11.7061	11.5435
21	6.91396	56.778	7.78363	12.6596	12.847
22	8.67137	56.2082	7.82607	10.8184	11.1689

23	11.9539	55.5367	10.0742	12.3079	12.3264
24	6.10968	57.9415	11.7409	14.9771	13.2923
25	11.0402	54.8729	7.98041	10.0039	10.325
26	5.49512	57.0916	7.69106	9.66118	11.0924
27	3.58995	60.107	14.3288	16.4371	15.918
28	8.44576	55.4546	8.71867	10.7879	10.0196
29	2.15994	60.3899	16.0717	18.4312	14.3238
30	5.742	57.5842	8.46733	11.9745	12.7213
Total	211,93632	1714,4299	280,62168	366,86902	359,34981
Rata	7,064544	57,147663	9,354056	12,228967	11,978327

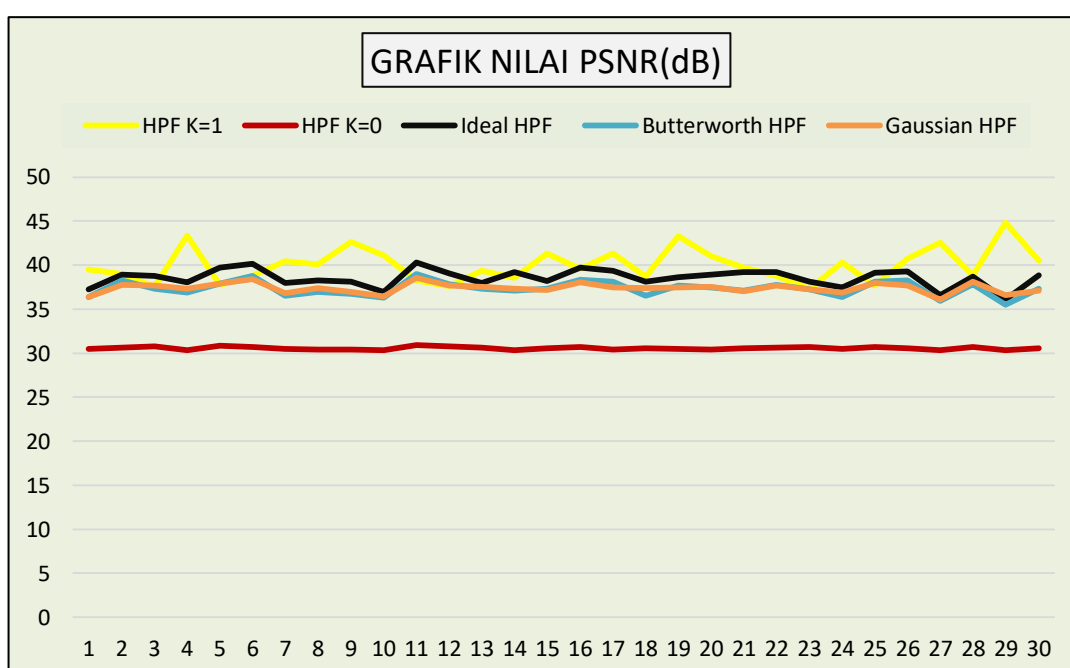


4.16. Grafik Nilai MSE

Tabel 4.6. Nilai PSNR(dB)

No	High Pass Filter		Fast Fourier Transform (D₀=5)		
	<i>HPF K=1</i>	<i>HPF K=0</i>	<i>Ideal HPF</i>	<i>Butterworth HPF(n=2)</i>	<i>Gaussian HPF</i>
1	39.495	30.5093	37.2707	36.3557	36.3731
2	38.9597	30.6652	38.9041	38.2495	37.763
3	37.5323	30.7595	38.7459	37.3208	37.6934
4	43.3507	30.3423	38.0585	36.8921	37.3224
5	37.7877	30.875	39.7299	37.8667	37.8885
6	38.7395	30.697	40.1469	38.7796	38.3883
7	40.4358	30.4502	37.9433	36.5019	36.7775
8	40.0969	30.4359	38.2594	36.961	37.3656
9	42.6138	30.4161	38.1116	36.7136	36.9442
10	41.096	30.335	36.9374	36.2751	36.4374
11	38.3	30.9062	40.3063	38.9798	38.5721
12	37.5799	30.7857	39.0606	37.8092	37.7031
13	39.3393	30.6225	37.9811	37.3368	37.5262
14	38.5355	30.3641	39.194	37.1212	37.3352
15	41.3361	30.5766	38.1789	37.3117	37.1634
16	39.5707	30.6736	39.7121	38.3054	38.0257
17	41.2936	30.439	39.3306	38.1441	37.4937
18	38.6866	30.5492	38.1195	36.5022	37.4178
19	43.2787	30.4744	38.6201	37.6832	37.4772
20	41.0538	30.4428	38.9536	37.4467	37.5074
21	39.7335	30.589	39.219	37.1066	37.0428
22	38.7499	30.6328	39.1954	37.7892	37.6507
23	37.3557	30.685	38.0987	37.229	37.2224
24	40.2706	30.5009	37.4338	36.3765	36.8948
25	37.701	30.7372	39.1106	38.1291	37.9919
26	40.731	30.5651	39.2709	38.2805	37.6806

27	42.5799	30.3416	36.5687	35.9726	36.1119
28	38.8644	30.6914	38.7263	37.8014	38.1223
29	44.7864	30.3212	36.0702	35.4753	36.5702
30	40.5402	30.5278	38.8533	37.3482	37.0855
Total	1200,3942	916,9116	1156,1114	1120,0647	1121,5483
Rata	40,01314	30,56372	38,537046	37,33549	37,384943



4.17. Grafik Nilai PSNR

4.2.3. Implementasi

Penerapan sistem ini digunakan dalam merekonstruksi citra telapak tangan dengan memperbaiki kualitasnya dengan menggunakan proses *image enhancement*. Penerapan *image enhancement* pada sistem ini dapat dilihat dari proses meningkatkan kontras pada citra telapak tangan agar citra tersebut menghasilkan kontras yang lebih tajam dengan menggunakan histogram ekualisasi. Kemudian citra yang sudah ditingkatkan kontrasnya di terapkan lagi proses *image enhancement* khususnya pada penajaman garis-garis telapak tangan citra dengan memakai metode *high pass-filter* dan *filter fast fourier transform*.

Sehingga tujuan sistem dalam penelitian ini adalah untuk membuat citra telapak tangan yang lebih baik, lebih jelas detail dan cirinya sehingga mudah diinterpretasikan oleh manusia serta citra telapak tangan yang telah di perbaiki kualitasnya yang nantinya dapat digunakan ketahap selanjutnya salah satunya seperti penerapan di bidang ilmu biometrik maupun bidang lainnya.

BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Dilihat dari penelitian yang dilakukan oleh penulis dalam memperbaiki kualitas citra telapak tangan dengan memanfaatkan teknik *high-pass filter* dan *fast fourier transform*, maka didapatkan penyelesaian dari penelitian ini adalah sebagai berikut:

1. Hasil penerapan metode *High-Pass Filter* terhadap citra telapak tangan yang telah di histogram ekualisasi, menunjukkan hasil dimana dengan $K=1$ menghasilkan citra rekonstruksi yang lebih baik dengan rata-rata nilai $MSE = 7,064544$ (dB) dan $PSNR = 40,01314$ (dB) dibanding dengan HPF dimana $K=0$ menunjukkan nilai rata-rata $MSE = 57,147663$ (dB) dan $PSNR = 30,56372$ (dB) .
2. Penerapan metode *Fast Fourier Transform* terhadap citra telapak tangan yang telah di histogram ekualisasi, menunjukkan hasil dimana dengan menggunakan *filter Ideal High Pass-Filter* (IHPF) menghasilkan citra rekonstruksi yang lebih baik dengan rata-rata nilai $MSE = 9,354056$ (dB) dan $PSNR = 38,537046$ (dB) dibanding dengan *filter Gaussian High Pass-Filter* (GHPF) dimana menunjukkan nilai rata-rata $MSE = 11,978327$ (dB) dan $PSNR = 37,384943$ (dB) dan *filter Butterworth High Pass-Filter* (BHPF) order 2 dengan hasil rerata $MSE = 12,228967$ (dB) serta nilai rerata $PSNR = 37,33549$ (dB).
3. Berdasarkan nilai MSE dan PSNR normal di atas, hal ini menunjukkan bahwa semakin rendah nilai MSE yang dibuat oleh gambar, semakin baik gambar yang dibuat ulang. Kemudian, jika nilai PSNR yang dihasilkan oleh gambar tersebut semakin besar, gambar yang dibuat ulang akan menunjukkan hasil yang bagus juga..
4. Penggunaan metode-metode dalam *image enhancement* seperti histogram ekualisasi, *high-pass filter* dan *filter fast fourier transform* mampu meningkatkan

kualitas citra menjadi lebih baik, detail, dan jelas sehingga mudah diinterpretasikan manusia.

5.2. Saran

Adapun saran-saran yang perlu disampaikan penulis demi perbaikan dan kemajuan sistem untuk apa yang akan datang adalah sebagai berikut:

1. Dalam penulisan maupun sistem yang di buat dalam penelitian ini, penulis mengingat bahwa masih banyak kesalahan-kesalahan yang harus dikoreksi.
2. Sistem yang penulis gunakan terhadap penelitian ini, masih mempunyai banyak kekurangan. Sehingga, diharapkan untuk kedepannya dapat dikembangkan lagi khususnya dalam penerapan metode-metode dalam *image enhancement* yang lebih luas lagi.

DAFTAR PUSTAKA


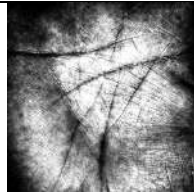
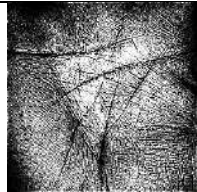







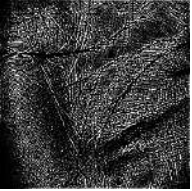



- Arif, R. B., Mahmudur, M., Khan, R., & Siddique, M. A. B. (2018). *Digital Image Enhancement in Matlab: An Overview on Histogram Equalization and Specification*.
- Ariyanto, M., & Caesarendra, W. (2011). *Panduan Belajar Mandiri MATLAB*.
- Chahyati, D. (2004). *Modul Perkuliahan Pengolahan Citra*.
- Dony, H., Putra, A., Isnanto, R. R., & Zahra, A. A. (2015). *Kinerja Pengenalan Sistem Biometrik Telapak Tangan Menggunakan Ekstraksi Ciri Berbasis Gelombang Singkat Biortogonal*. 4.
- Fahmi, F. (2013). Klasifikasi Dan Peningkatan Kualitas Citra Sidik Jari Menggunakan FFT (Fast Fourier Transform). *Singuda Ensikom*, 6–11.
- Ferrarese, F. P. (2002). *IMAGE ENHANCEMENT: Filtering In The Frequency Domain*.
- Furqan, M., Sriani, S., & Siregar, Y. K. (2020). Perbandingan Algoritma Contraharmonic Mean Filter dan Arithmetic Mean Filter untuk Mereduksi Exponential Noise. *JISKA (Jurnal Informatika Sunan Kalijaga)*, 5(2), 107.
- G.Khandizod, A., & Deshmukh, R. R. (2015). Comparative Analysis of Image Enhancement Technique for Hyperspectral Palmprint Images. *International Journal of Computer Applications*, 121(23), 30–35.
- Gonzalez, R. C. (2002). *Digital_Image_Processing_2ndEd.pdf* (pp. 1–793).
- Imron, M. (2013). *Makalah Pengolahan Citra*.
- Munir, R. (2004). *Pengolahan Citra Digital Dengan Pendekatan Algoritmik*.
- Palanikumar, S., Sasikumar, M., & Rajeesh, J. (2011). Palmprint Enhancement Using Discrete Curvelet Transform. *International Journal of Computer Science Issues*, 8(4), 313–319.
- Prakasa, E., & Herlan, H. (2017). *Pengujian Klasifikasi Pola Telapak Tangan dengan Menggunakan Rangkaian Matriks Sensor Cahaya*.
- Prasetyo, E. (2011). *Pengolahan dalam Domain Frekuensi dan Restorasi Citra*.




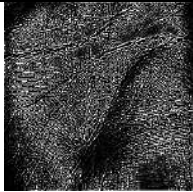
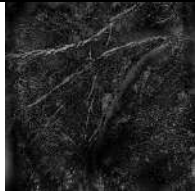
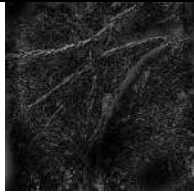
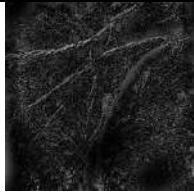
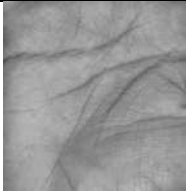


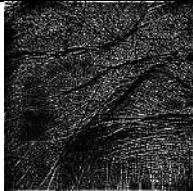




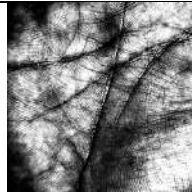
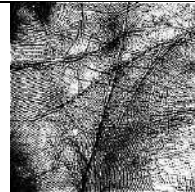
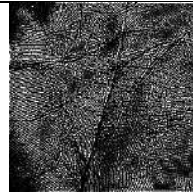





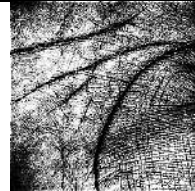




- Priyawati, D. (2011). Teknik Pengolahan Citra Digital Berdomain Spasial Untuk Peningkatan Citra Sinar-X. *KomuniTi, II*, 44–50.
- Purnama, B., & Salsabila, D. (2013). *Identifikasi Telapak Tangan Menggunakan 2DPCA plus PCA dan K- Nearest Neighbour*. 1–6.
- Rahman, S. (2017). Perancangan Aplikasi Identifikasi Biometrika Telapak Tangan Menggunakan Metode Freeman Chain Code. *CESS*, 2(2), 64–73.
- Rahmawati, E., Wijianto, R., & Riana, D. (2018). *Perbandingan Metode Peningkatan Kualitas Citra dan Deteksi Tepi pada Citra Kutu Kebul*.
- Renaningtias, N., Efendi, R., & Susilo, B. (2015). *Aplikasi Biometrika Pencocokan Citra Garis Telapak Tangan Dengan Metode Transformasi Wavelet Dan Mahalonibis Distance*. 3(2), 95–106.
- Rupa, B., Karuna, G., & Reddy, G. V. R. (2018). A Survey on Spatial Domain Image Enhancement Techniques. *IJRASET*, 6(IV).
- Setiawan, R. A., Isnanto, R. R., & Hidayatno, A. (2013). *Identifikasi Diri Berdasarkan Biometrika Telapak Tangan Menggunakan Metode Tapis Gabor 2-D dan Jaringan Syaraf Tiruan Learning Vector Quantization (LVQ)*.
- Singh, L. S. S., Ahlawat, A. K., Singh, K. M., & Singh, T. R. (2017). A Review on Image Enhancement Methods on Different Domains. *International Journal of Engineering Inventions*, 6(1), 49–55.
- Sipasulta, R. Y., St, A. S. M. L., & Sompie, S. R. U. A. (2014). Simulasi Sistem Pengacak Sinyal Dengan Metode FFT (Fast Fourier Transform). *E-Journal Teknik Elektro Dan Komputer*, 3(2), 1–9.
- Siregar, A., & Aryanta, D. W. I. (2013). Simulasi dan Analisis Perbaikan Citra Digital Domain Frekuensi dengan Transformasi Fourier. *Reka Elkomika*, 1(3), 198–209.
- Sri Arsa et al. (2017). Comparison Of Image Enhancement Methods For Chromosome Karyotype Image Enhancement. *Ilmu Komputer Dan Informasi*, 1, 50–58.
- Sriani, Triase, & Khairuna. (2017). Pendekomposisian Citra Digital Dengan Algoritma DWT. *Jurnal Ilmu Komputer Dan Informatika*, 01(01), 35–39.



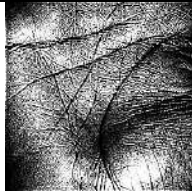






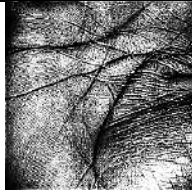
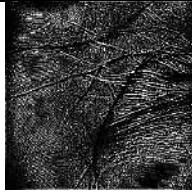




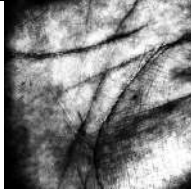
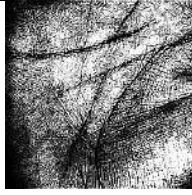






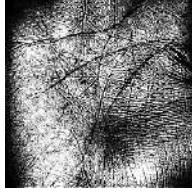
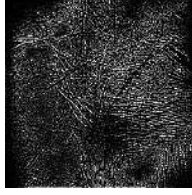



LAMPIRAN-LAMPIRAN



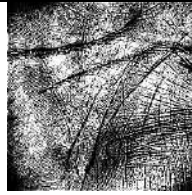













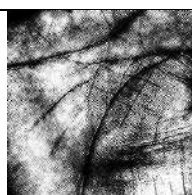






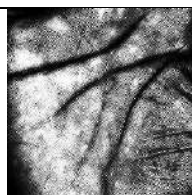
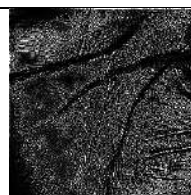



Lampiran 1

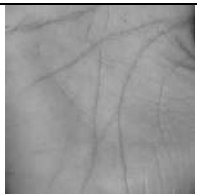


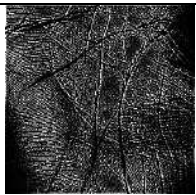



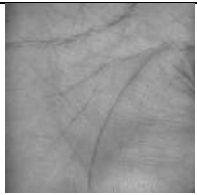


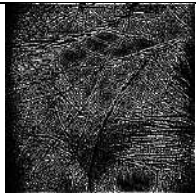



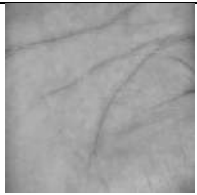


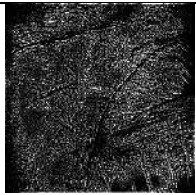



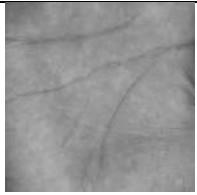


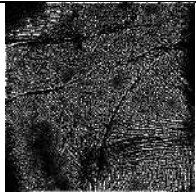

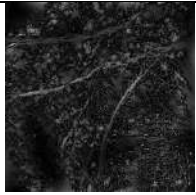

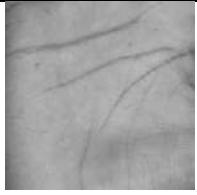
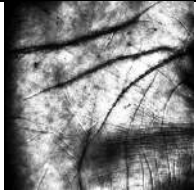





Hasil uji citra telapak tangan pada “*Image Enhancement Menggunakan Metode High Pass Filter Dan Fast Fourier Transform Pada Citra Telapak Tangan*”:

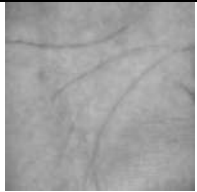
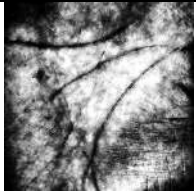
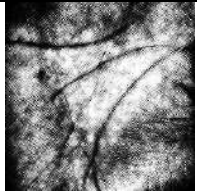
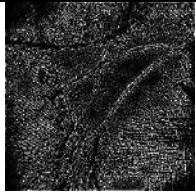
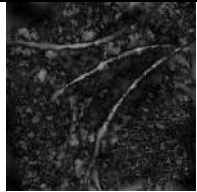


















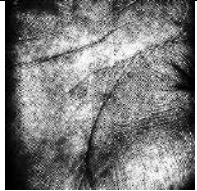
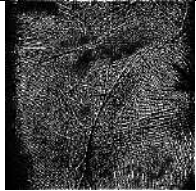


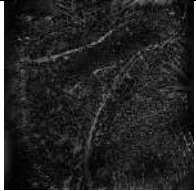

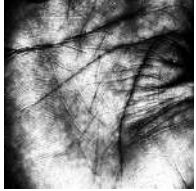


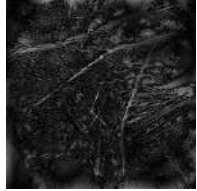


No	Grayscale	Histogram Ekualisasi	<i>High-Pass Filter</i>		<i>Fast Fourier Transform (D₀=5)</i>		
			HPF K=1	HPF K=0	IHPF	BHPF (n=2)	GHPF
1							
2							




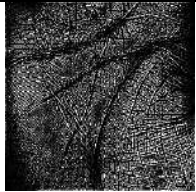





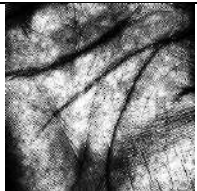
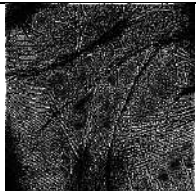





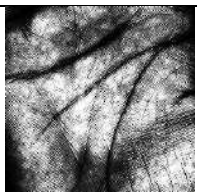




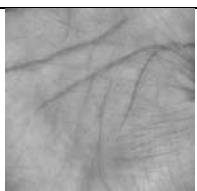

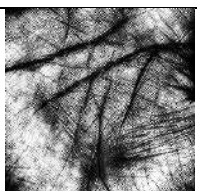
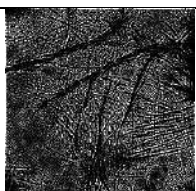




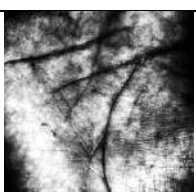


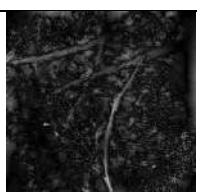
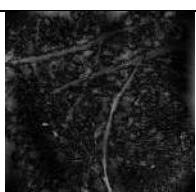
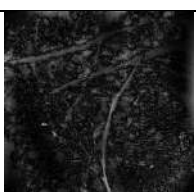
3							
4							
5							
6							

7							
8							
9							
10							

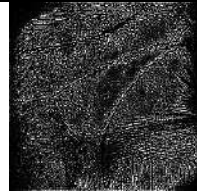
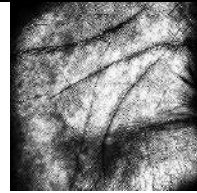
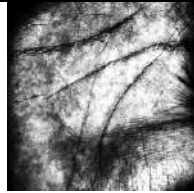
11							
12							
13							
14							

15							
16							
17							
18							
19							

20							
21							
22							
23							
24							



25							
26							
27							
28							
29							



30




Lampiran 2

Alat bantu (*tools*) yang digunakan pada Matlab GUI untuk “*Image Enhancement Menggunakan Metode High-Pass Filter Dan Fast Fourier Transform Pada Citra Telapak Tangan*”:

UI Control	Tag	Fungsi
 Push Button	Tombol_Browse	Mengambil citra di folder penyimpanan dan langsung di <i>resize</i> dan <i>grayscale</i>
	Tombol_Histeq	Melakukan histogram ekualisasi pada citra
	Tombol_Reset	Mereset ulang seperti semula
	Simpan_Histeq	Menyimpan citra hasil histogram ekualisasi
	Tombol_Hpf1	Melakukan filter HPF K=1
	Tombol_Hpf0	Melakukan filter HPF K=0
	Hitung_HPF	Menghitung MSE dan PSNR dari citra HPF
	Simpan_K1	Menyimpan citra hasil HPF K=1
	Simpan_K0	Menyimpan citra hasil HPF K=0
	Tombol_FFT	Melakukan filter FFT (IHPF,BHPF,GHPF)
	Hitung_FFT	Menghitung MSE dan PSNR dari citra filter FFT
	Simpan_IHPF	Menyimpan citra hasil filter IHPF
	Simpan_BHPF	Menyimpan citra hasil filter BHPF
	Simpan_GHPF	Menyimpan citra hasil filter GHPF
 Statistic Text	Judul	Menampilkan teks “Judul”
	x1	Menampilkan simbol “x” dari axes ori
	x2	Menampilkan simbol “x” dari axes gray
	Preview	Menampilkan teks “Pre-Processing Preview”
	Do	Menampilkan teks “D0”
	MSE_k1	Menampilkan teks “MSE” pada HPF K=1
	PSNR_k1	Menampilkan teks “PSNR” pada HPF K=1
	MSE_k0	Menampilkan teks “MSE” pada HPF K=0
	PSNR_k0	Menampilkan teks “PSNR” pada HPF K=0

	MSE_ih	Menampilkan teks “MSE” pada IHPF
	PSNR_ih	Menampilkan teks “PSNR” pada IHPF
	MSE_bh	Menampilkan teks “MSE” pada BHPF
	PSNR_bh	Menampilkan teks “PSNR” pada BHPF
	MSE_gh	Menampilkan teks “MSE” pada GHPF
	PSNR_gh	Menampilkan teks “PSNR” pada BHPF
 Edit Text	Baris1	Menampilkan resolusi “baris” dari citra ori
	Kolom1	Menampilkan resolusi “kolom” dari citra ori
	Baris2	Menampilkan resolusi “baris” dari citra <i>gray</i>
	Kolom1	Menampilkan resolusi “kolom” dari citra <i>gray</i>
	edit_MSE1	Menampilkan nilai MSE dari HPF K=1
	edit_PSNR1	Menampilkan nilai PSNR dari HPF K=1
	edit_MSE0	Menampilkan nilai MSE dari HPF K=0
	edit_PSNR0	Menampilkan nilai PSNR dari HPF K=0
	edit_D0	Menginput nilai <i>cutoff</i> pada <i>filter</i> FFT
	MSE_IHPF	Menampilkan nilai MSE dari IHPF
	PSNR_IHPF	Menampilkan nilai PSNR dari IHPF
	MSE_BHPF	Menampilkan nilai MSE dari BHPF
	PSNR_BHPF	Menampilkan nilai PSNR dari BHPF
	MSE_GHPF	Menampilkan nilai MSE dari GHPF
	PSNR_GHPF	Menampilkan nilai PSNR dari GHPF
 Axes	axes_Ori	Menampilkan citra original hasil input citra
	axes_Gray	Menampilkan citra hasil <i>grayscale</i>
	axes_Histogram	Menampilkan histogram dari citra his. Ekualisasi
	axes_Ekualisasi	Menampilkan citra histogram ekualisasi
	axes_Preview	Menampilkan citra-citra hasil <i>pre-processing</i>
	axes_HPF1	Menampilkan citra hasil <i>filter</i> HPF K=1
	axes_HPF0	Menampilkan citra hasil <i>filter</i> HPF K=0
	axes_IHPF	Menampilkan citra hasil <i>filter</i> IHPF
	axes_BHPF	Menampilkan citra hasil <i>filter</i> BHPF

	axes_GHPF	Menampilkan citra hasil <i>filter</i> GHPF
 <i>Panel</i>	Prepro_Panel	Sebagai panel/kontainer untuk <i>pre-processing</i>
	HPF_Panel	Sebagai panel/kontainer untuk <i>High-Pass Filter</i>
	FFT_Panel	Sebagai panel/kontainer untuk <i>Fast Fourier Transform</i>
	ID_MHS	Sebagai panel menampilkan Identitas penulis/mahasiswa

Lampiran 3

Source code dari program Matlab “Image Enhancement Menggunakan Metode High-Pass Filter Dan Fast Fourier Transform Pada Citra Telapak Tangan”:

```
function varargout = His1(varargin)
% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @His1_OpeningFcn, ...
                  ...
                  'gui_OutputFcn',  @His1_OutputFcn, ...
                  ...
                  'gui_LayoutFcn',   [] , ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State,
varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

% End initialization code - DO NOT EDIT
function Aplikasi_Enhancement_OpeningFcn(hObject, ~,
handles, varargin)
handles.output = hObject;
guidata(hObject, handles);

% ==> Membuat Latar Belakang Sistem
ah=axes ('unit','normalized','position',[0 0 1 1]);
uistack(ah,'bottom');
[back, map] = imread('BgColor.jpg');
image(back);
colormap (map)
set (ah, 'handlevisibility','off','visible','off')

function varargout = His1_OutputFcn(~, ~, handles)
varargout{1} = handles.output;
function Tombol_Browse_Callback(hObject, ~, handles)
```

```

% ==>Memasukkan File Citra
[filename,pathname]=uigetfile({'*.bmp','File
BMP(*.bmp)'),'Buka File Citra');
handles.image_file = strcat(pathname,filename);
if exist(handles.image_file)
handles.Citra_Ori=imread(handles.image_file);
cla(handles.axes_Ori,'reset');
axes(handles.axes_Ori);
imshow(handles.Citra_Ori);
else                                     % Jika Citra Tidak
Ada
h = msgbox('File not found!','Warning');    %Tampil
Pesannya
end
hold on;
guidata (hObject,handles);

% ==>Meresize Citra Original
Citra_Ori = handles.Citra_Ori;
Citra_Resize = imresize(Citra_Ori, [256 256]);
handles.Tombol_Histeq = Citra_Resize;
guidata(hObject,handles);
cla(handles.axes_Gray,'reset');
axes(handles.axes_Gray);

% ==>Menampilkan Ukuran Resolusi Citra Sebelum dan
Sesudah Resize
set(handles.Baris1,'string',size(handles.Citra_Ori,1));
set(handles.Kolom1,'string',size(handles.Citra_Ori,2));
set(handles.Baris2,'string',size(handles.Tombol_Histeq,
1));
set(handles.Kolom2,'string',size(handles.Tombol_Histeq,
2));

% ==>Mengubah Citra RGB ke Citra Grayscale
Citra_Resize = handles.Citra_Ori;
Citra_Gray = rgb2gray(Citra_Resize);
axes(handles.axes_Gray)
imshow(Citra_Gray)

% ==> Menampilkan Histogram dari Citra Grayscale
axes(handles.axes_Histogram)
imhist(Citra_Gray);

% ==>Menampilkan Citra Grayscale ke Axes Preview
axes(handles.axes_Preview)
imshow(Citra_Gray);

```

```

function Baris1_Callback(~, ~, ~)
function Baris1_CreateFcn(hObject, ~, ~)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

function Kolom1_Callback(~, ~, ~)
function Kolom1_CreateFcn(hObject, ~, ~)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

function Baris2_Callback(~, ~, ~)
function Baris2_CreateFcn(hObject, ~, ~)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function Kolom2_Callback(~, ~, ~)
function Kolom2_CreateFcn(hObject, ~, ~)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

function Tombol_Histeq_Callback(~, ~, handles)
% ==> Melakukan Proses Histogram Ekualisasi
Citra_Ori = handles.Citra_Ori;
Citra_Resize = imresize(Citra_Ori, [256 256]);
handles.Citra_Resize = Citra_Resize;
New_Gray=rgb2gray(uint8(Citra_Resize));
cla(handles.axes_Ekualisasi,'reset');
axes(handles.axes_Ekualisasi);
histeq(New_Gray);
axes(handles.axes_Preview);
histeq(New_Gray);
hold on;
% ==> Mendapatkan Histogram dari Citra Histogram
Ekualisasi
Histo_Citra=histeq(New_Gray);
cla(handles.axes_Histogram,'reset');
axes(handles.axes_Histogram);
imhist(Histo_Citra);

```

```

set(gca,'XTickLabelMode','auto'); %Tampilkan label X-
axis
hold on;

function Simpan_Histeq_Callback(~, ~, handles)
% ==> Menyimpan Citra Hasil Pre-Processing
[nama_file_simpan,path_simpan]=uiputfile(...
{'*.bmp'},'Simpan Citra Pre-Processing');
Save_Prepro=getimage(handles.axes_Ekualisasi
imwrite(Save_Prepro,fullfile(path_simpan,nama_file_simp
an),'bmp');

function Tombol_Hpf1_Callback(~, ~, handles)
% ==> Melakukan Konvolusi Dengan Filter HPF K=1
hpf=getimage(handles.axes_Preview);
hpf1=[ 1 -2 1;-2 5 -2;1 -2 1];
kernel1=uint8(conv2(double(hpf),hpf1,'same'));
axes(handles.axes_HPF1)
imshow(kernel1);title('K=1')

function Tombol_Hpf0_Callback(~, ~, handles)
% ==> Konvolusi Filter HPF K=0
hpf=getimage(handles.axes_Preview);
hpf0=[-1 -1 -1;-1 8 -1;-1 -1 -1];
kernel0=uint8(conv2(double(hpf),hpf0,'same'));
axes(handles.axes_HPF0)
imshow(kernel0);title('K=0')

function Simpan_K1_Callback(~, ~, handles)
% ==> Menyimpan Citra HPF K=1
[nama_file_simpan,path_simpan]=uiputfile(...
{'*.bmp'},'Simpan Citra HPF');
Save_HPF=getimage(handles.axes_HPF1);
imwrite(Save_HPF,fullfile(path_simpan,nama_file_simpan)
,'bmp');

```

```

function Simpan_K0_Callback(~, ~, handles)
% ==> Menyimpan Citra HPF K=0
[nama_file_simpan,path_simpan]=uiputfile(...
    {'*.bmp'}, 'Simpan Citra HPF');
Save_HPF=getimage(handles.axes_HPF0);
imwrite(Save_HPF,fullfile(path_simpan,nama_file_simpan)
, 'bmp');

function Tombol_FFT_Callback(~, ~, handles)
% ==> Melakukan Proses Enhancement dengan Filter Fast
Fourier Transform
FFT_ihpf=getimage(handles.axes_Preview); %Mengambil
Citra dari Axes
[M, N]=size(FFT_ihpf); %Menyimpan Ukuran
Citra
f_transform=fft2(FFT_ihpf); %Mendapatkan FT dari
Citra
f_shift=fftshift(f_transform); %Ubah ke Center spectrum
f_log=log(1+abs(f_shift)); %Ubah C.Spectrum ke Log
Transform

%Menampilkan FT, Center Spectrum dan Log Transform
IHPF, BHPF,GHPH
figure(2);
subplot(3,4,1);imshow(abs(f_transform),[]); title('FT
of Image')
subplot(3,4,2);imshow(abs(f_shift),[]); title('Center
Spectrum Image')
subplot(3,4,3);imshow(abs(f_log),[]); title('Log
Transform Image')
subplot(3,4,5);imshow(abs(f_transform),[]); title('FT
of Image')
subplot(3,4,6);imshow(abs(f_shift),[]); title('Center
Spectrum Image')
subplot(3,4,7);imshow(abs(f_log),[]); title('Log
Transform Image')
subplot(3,4,9);imshow(abs(f_transform),[]); title('FT
of Image')
subplot(3,4,10);imshow(abs(f_shift),[]); title('Center
Spectrum Image')
subplot(3,4,11);imshow(abs(f_log),[]); title('Log
Transform Image')

n=2; %Orde Untuk Filter BHPF
D0=str2double(get(handles.edit_D0, 'String')); %Input
Nilai Cutoff

```

```

%Perancangan Filter
u=M/2;
v=N/2;
for i=1:M
for j=1:N
    %Menghitung Jarak D(u,v)
    Distance=sqrt((i-u).^2+(j-v).^2);

    %Menentukan Kernel Filter
    Kernel_IHPF(i,j)= double(Distance > D0);%IHPF
    Kernel_BHPF(i,j)= 1./(1 +
(D0./Distance).^(2*n));%BHPF
    Kernel_GHPF(i,j)=1-exp(-
(Distance)^2/(2*(D0^2)));%GHPF
end
end

%Perkalian Transformasi Fourier dan Kernel Filter
IHPF= f_shift .* Kernel_IHPF;
BHPF= f_shift .* Kernel_BHPF;
GHPF= f_shift .* Kernel_GHPF;

%Invers FFT Shift
IHPF_Shift=ifftshift(IHPF);
BHPF_Shift=ifftshift(BHPF);
GHPF_Shift=ifftshift(GHPF);

%Subplot Untuk Menampilkan Invers FFT Shift
subplot(3,4,4);imshow(abs(log(IHPF_Shift)),[]);
title('Convolution IHPF')
subplot(3,4,8);imshow(abs(log(BHPF_Shift)),[]);
title('Convolution BHPF')
subplot(3,4,12);imshow(abs(log(GHPF_Shift)),[]);
title('Convolution GHPF')

%Invers Fast Fourier Transform
Ori_IHPF=abs(ifft2(IHPF_Shift));
Ori_BHPF=abs(ifft2(BHPF_Shift));
Ori_GHPF=abs(ifft2(GHPF_Shift));

%Menampilkan Citra Output ke Dalam Axes
axes(handles.axes_IHPF);imshow(abs(uint8(Ori_IHPF)),[]);
; title('Citra IHPF')
axes(handles.axes_BHPF);imshow(abs(uint8(Ori_BHPF)),[]);
; title('Citra BHPF')

```



```
axes(handles.axes_GHPF);imshow(abs(uint8(ori_GHPF)),[])
; title('Citra GHPF')
```

```
function edit_D0_Callback(~, ~, ~)
function edit_D0_CreateFcn(hObject, ~, ~)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function Hitung_HPF_Callback(~, ~, handles)
% ==> SEBELUM MEMASUKKAN SOURCE CODE HITUNG MSE dan
PSNR,TERLEBIH DAHULU BUAT M-FILE UNTUK FUNCTION RUMUS
MSE dan PSNR.
```

```
%DIBAWAH INI ADALAH FUNCTION DARI NILAI MSE DAN PSNR
```

```
function [nilaiMSE, nilaiPSNR] =
MSE_PSNR(citra_Prepro,citra_Process)
```

```
%MSE
```

```
[m, n] = size(double(citra_Process));
nilaiMSE = sum(sum(citra_Prepro - citra_Process
.^2))/(m*n);
```

```
%PSNR
```

```
nilaiPSNR = 10*log10(255*255/(nilaiMSE));
```

```
end
```

```
% ==> Lalu Callback Tombol Hitung Masukkan Kode Berikut
```

```
% ==> Menampilkan Nilai MSE dan PSNR
```

```
citra_Prepro= getimage(handles.axes_Preview);
citra_Process= getimage(handles.axes_HPF1);
[nilaiMSE, nilaiPSNR]=
MSE_PSNR(citra_Prepro,citra_Process);
set(handles.edit_MSE1,'String', nilaiMSE);
set(handles.edit_PSNR1,'String', nilaiPSNR);
```

```
citra_Prepro= getimage(handles.axes_Preview);
citra_Process= getimage(handles.axes_HPF0);
[nilaiMSE, nilaiPSNR]=
MSE_PSNR(citra_Prepro,citra_Process);
set(handles.edit_MSE0,'String', nilaiMSE);
set(handles.edit_PSNR0,'String', nilaiPSNR);
```

```
function edit_MSE1_Callback(~, ~, ~)
function edit_MSE1_CreateFcn(hObject, ~, ~)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
```

end

```
function edit_PSNR1_Callback(~, ~, ~)
function edit_PSNR1_CreateFcn(hObject, ~, ~)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function edit_MSE0_Callback(~, ~, ~)
function edit_MSE0_CreateFcn(hObject, ~, ~)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function edit_PSNR0_Callback(~, ~, ~)
function edit_PSNR0_CreateFcn(hObject, ~, ~)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function Hitung_FFT_Callback(~, ~, handles)
% ==> Menghitung Nilai MSE dan PSNR dari Citra IHPF
citra_Prepro= getimage(handles.axes_Preview);
citra_Process= getimage(handles.axes_IHPF);
[nilaiMSE, nilaiPSNR]=
MSE_PSNR(citra_Prepro,citra_Process);
set(handles.MSE_IHPF,'String', nilaiMSE);
set(handles.PSNR_IHPF,'String', nilaiPSNR);
```

```
% ==> Menghitung Nilai MSE dan PSNR dari Citra BHPF
citra_Prepro= getimage(handles.axes_Preview);
citra_Process= getimage(handles.axes_BHPF);
[nilaiMSE, nilaiPSNR]=
MSE_PSNR(citra_Prepro,citra_Process);
set(handles.MSE_BHPF,'String', nilaiMSE);
set(handles.PSNR_BHPF,'String', nilaiPSNR);
% ==> Menghitung Nilai MSE dan PSNR dari Citra GHPF
citra_Prepro= getimage(handles.axes_Preview);
citra_Process= getimage(handles.axes_GHPF);
[nilaiMSE, nilaiPSNR]=
MSE_PSNR(citra_Prepro,citra_Process);
set(handles.MSE_GHPF,'String', nilaiMSE);
set(handles.PSNR_GHPF,'String', nilaiPSNR);
```

```

function MSE_IHPF_Callback(hObject, eventdata, handles)
function MSE_IHPF_CreateFcn(hObject, ~, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function PSNR_IHPF_Callback(hObject, eventdata,
handles)
function PSNR_IHPF_CreateFcn(hObject, ~, ~)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function MSE_BHPF_Callback(~, ~, handles)
function MSE_BHPF_CreateFcn(hObject, ~, ~)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function PSNR_BHPF_Callback(~, ~, ~)
function PSNR_BHPF_CreateFcn(hObject, ~, ~)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function MSE_GHPF_Callback(~, ~, ~)
function MSE_GHPF_CreateFcn(hObject, ~, ~)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function PSNR_GHPF_Callback(~, ~, ~)
function PSNR_GHPF_CreateFcn(hObject, ~, ~)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function Simpan_IHPF_Callback(~, ~, handles)
% ==> Menyimpan Citra IHPF
[nama_file_simpan,path_simpan]=uiputfile(...

```

```

        {'*.bmp'}, 'Simpan Citra IHPF');
Save_HPF=getimage(handles.axes_IHPF);
imwrite(Save_HPF,fullfile(path_simpan,nama_file_simpan)
,'bmp');

```

```

function Simpan_BHPF_Callback(~, ~, handles)
% ==> Menyimpan Citra BHPF
[nama_file_simpan,path_simpan]=uiputfile(...
    {'*.bmp'}, 'Simpan Citra BHPF');
Save_HPF=getimage(handles.axes_BHPF);
imwrite(Save_HPF,fullfile(path_simpan,nama_file_simpan)
,'bmp');

```

```

function Simpan_GHPF_Callback(~, ~, handles)
% ==> Menyimpan Citra GHPF
[nama_file_simpan,path_simpan]=uiputfile(...
    {'*.bmp'}, 'Simpan Citra GHPF');
Save_HPF=getimage(handles.axes_BHPF);
imwrite(Save_HPF,fullfile(path_simpan,nama_file_simpan)
,'bmp');

```

```

function Tombol_Reset_Callback(~, ~, handles)
% ==> Mereset Ulang Seperti Semula
set (handles.Baris1,'string','');
set (handles.Kolom1,'string','');
set (handles.Baris2,'string','');
set (handles.Kolom2,'string','');
set (handles.edit_MSE1,'string','');
set (handles.edit_PSNR1,'string','');
set (handles.edit_MSE0,'string','');
set (handles.edit_PSNR0,'string','');
set (handles.edit_D0,'string','');
set (handles.MSE_IHPF,'string','');
set (handles.PSNR_IHPF,'string','');
set (handles.MSE_BHPF,'string','');
set (handles.PSNR_BHPF,'string','');
set (handles.MSE_GHPF,'string','');
set (handles.PSNR_GHPF,'string','');
axes (handles.axes_Ori);cla('reset');
axes (handles.axes_Gray);cla('reset');
axes (handles.axes_Histogram);cla('reset');
axes (handles.axes_Ekualisasi);cla('reset');
axes (handles.axes_Preview);cla('reset');
axes (handles.axes_HPF1);cla('reset');
axes (handles.axes_HPF0);cla('reset');
axes (handles.axes_IHPF);cla('reset');
axes (handles.axes_BHPF);cla('reset');

```


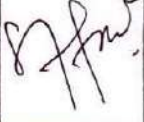

```
axes(handles.axes_GHPF);cla('reset');  
close (figure(2))  
clc;  
clc;  
clc;  
clc;  
clc;  
clc;  
clear all;
```



KARTU BIMBINGAN SKRIPSI
PROGRAM STUDI ILMU KOMPUTER
FAKULTAS SAINS DAN TEKNOLOGI

T.A 2019/2020

Nama : Mhd. Akbar Ramadhan T	Pembimbing I : Dr. Mhd. Furqan, S.Si., M.Comp. Sc
NIM : 0701163110	Pembimbing II : Sriani, S.Kom., M.Kom
Prog. Studi : Ilmu Komputer	
Judul Skripsi : ANALISIS PERBANDINGAN IMAGE ENHANCEMENT MENGGUNAKAN METODE HIGH-PASS FILTER (HPF) DAN FAST FOURIER TRANSFORM (FFT) PADA CITRA TELAPAK TANGAN	

PERT.	Tanggal	Materi Bimbingan	Tanda Tangan Pembimbing Skripsi	
			PS. I	PS.II
I	07 Juni 2020	Revisi BAB I (Terkait mengenai isi dari Latar Belakang, Rumusan Masalah, Batasan Masalah, dan Tujuan Penelitian)		
II	15 Juni 2020	Revisi BAB I (Rumusan masalah dan Tujuan Penelitian, dimana pada tujuan penelitian merupakan jawaban dari rumusan masalah).		
III	30 Juni 2020	ACC BAB I (Lanjut penyusunan BAB II, III, dan Daftar Pustaka)		
IV	03 Juli 2020	Revisi Judul Proposal Skripsi		
V	08 November 2020	Revisi BAB II (Susunan teori-teori terkait harus runtut, dan simbol-simbol Flowchart harus dibuat sendiri)		

VI	09 November 2020	Revisi BAB III (Bentuk Jadwal dan Waktu Penelitian dibuat sederhana, dan bentuk flowchart harus sesuai dengan diagram penelitian)	<i>Furqa</i>	<i>87 Hris</i>
VII	11 November 2020	ACC BAB I, II, III dan Daftar pustaka dari pembimbing II. Lanjut ke pembimbing I dengan melengkapi proposal skripsi	<i>Furqa</i>	<i>87 Hris</i>
VIII	16 November 2020	Bimbingan serta ACC Proposal Skripsi dari pembimbing I. Lanjut ke Seminar Proposal	<i>Furqa</i>	<i>87 Hris</i>
IX	18 Desember 2020	Revisi Proposal Skripsi Lanjut ke BAB IV dan V.	<i>Furqa</i>	<i>87 Hris</i>
X	3 Maret 2021	ACC BAB IV dan V Lanjut ke Abstrak, Daftar Pustaka, Lembar Persetujuan, Lampiran	<i>Furqa</i>	<i>87 Hris</i>
XI	16 Maret 2021	ABSTRAK, DAFTAR PUSTAKA, LEMBAR PERSETUJUAN, KESIMPULAN DAN LAMPIRAN	<i>Furqa</i>	<i>87 Hris</i>
XII	16 Maret 2021	ACC SIDANG MUNAQASYAH	<i>Furqa</i>	<i>87 Hris</i>

Medan, 17 Maret 2021

An. Dekan
Ketua Jurusan/Program Studi
Ilmu Komputer

Ilka Fufria S.Kom., M.Kom
NIP. 198306042015031006

RIWAYAT HIDUP



Muhammad Akbar Ramadhan Tanjung sebagai penulis tugas akhir ini. Penulis sendiri merupakan anak keempat dari lima bersaudara, sebagai putra kandung dari pasangan Bapak Dahlan dan Ibu Ernawati. Penulis yang dilahirkan di Desa Empat Negeri Kecamatan Lima Puluh, tepat pada tanggal 29 Januari 1997. Dan saat ini penulis tinggal bersama keluarga di kampung yang beralamat di Dusun VII RT/RW (07/07) Desa Empat Negeri, Kecamatan Lima Puluh, Batubara. Penulis dibesarkan dari keluarga yang pekerja keras dan sederhana serta menempuh pendidikan di kampung halaman. Dimulai dari Sekolah Dasar (SD) di SDN 010211 Simpang Dolok selama enam tahun dan lulus di tahun 2009. Melanjutkan pendidikan tingkat SLTP di SMP Negeri 2 Lima Puluh dan lulus pada tahun 2012. Kemudian melanjutkan ke jenjang SLTA di SMK Negeri 1 Lima Puluh, dan mengambil jurusan Multimedia serta lulus pada tahun 2015. Tepat pada Agustus 2016, penulis lulus ujian Mandiri di Universitas Islam Negeri Sumatera Utara (UINSU) Medan. Hingga akhirnya bisa menempuh masa kuliah di Jurusan Ilmu Komputer, Fakultas Sains Dan Teknologi UINSU Medan. Saat menempuh pendidikan perguruan tinggi, penulis melakukan Kuliah Kerja Nyata (KKN) di desa Teluk, Kecamatan Secanggang Kabupaten Langkat di tahun 2019. Dan Kerja Praktek (KP) di PT. Bank BNI Syariah Medan ditahun 2019.

Dengan ketekunan, serta motivasi yang tinggi agar terus berusaha dan belajar. Akhirnya penulis mampu menyelesaikan pengerjaan tugas akhir ini. Semoga dengan penulisan tugas akhir ini, diharapkan memberikan dampak positif dan berkontribusi bagi perkembangan dunia pendidikan.

Akhir kata penulis ucapkan rasa syukur yang sebesar-besarnya atas terselesaikannya skripsi ini yang berjudul “**Image Enhancement Menggunakan Metode High-Pass Filter Dan Fast Fourier Transform Pada Citra Telapak Tangan**”.

Lain-Lain

Nama Panggilan	: Akbar
NIM	: 0701163110
Fakultas/Jurusan	: Sains & Teknologi/ Ilmu komputer
Alamat Fakultas	: Jln. IAIN No. 1 Medan Sutomo 20371
E-mail	: mhdakbarramadhan@uinsu.ac.id
Agama	: Islam
Kebangsaan	: Indonesia
Tinggi/Berat Badan	: 166 cm/63 kg.